

به نام خدا

جزوه درسی مربوط به درس

بازی سازی

Game Making

حمید رضا نیرومند

<http://niroomand.ir>

انواع برنامه‌نویسی بازی

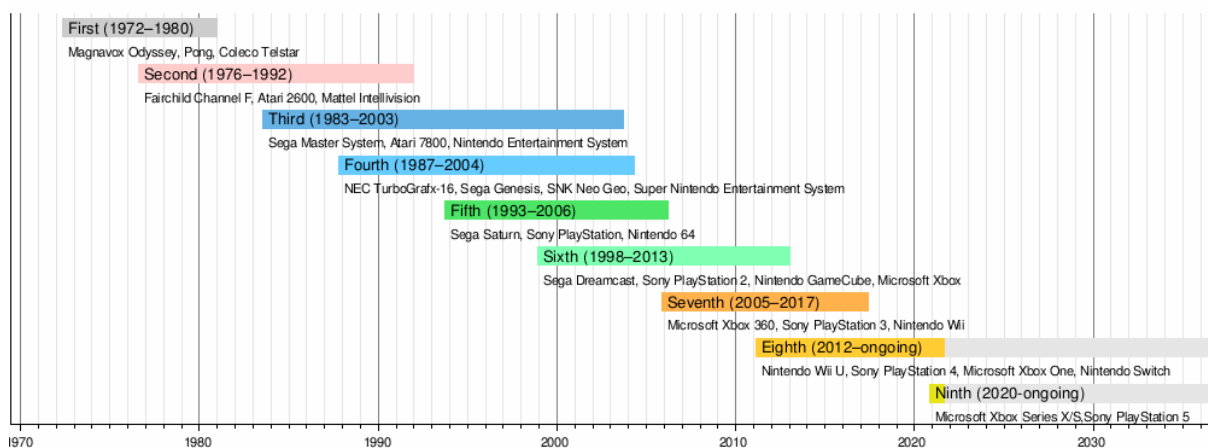
- 1 - ساخت بازی با یک زبان برنامه‌نویسی پایه مانند C++
- مختص شرکت‌های بزرگ بازی‌سازی
- 2 - ساخت بازی با کمک برنامه‌های کمکی مانند Unity
- قابل استفاده توسط شرکت‌های SME (کوچک و متوسط)
- 3 - ساخت بازی با زبان‌های تحت وب

مشهورترین شرکت‌های بازی‌سازی دنیا و مشهورترین بازی‌های آنها

- 1 - Nintendo
i. Mario
ii. Pokémon
iii. The legend of Zelda
- 2 - Electronic Arts
i. Battlefield series
ii. FIFA series
iii. The Simpsons series
iv. Star Wars series
- 3 - Activision Blizzard
i. Call of duty series
ii. Crash Bandicoot series
iii. Spyro the dragon series
- 4 - Ubisoft
i. Assassin's creed series
ii. Far cry series
- 5 - Naughty Dog Inc
i. The Last of Us
ii. Uncharted
- 6 - Epic Games
i. Fortnite
ii. Gears of War
iii. Infinity Blade
- 7 - Infinity Ward

Call of Duty .i	
Gameloft	- 8
Asphalt .i	
Brothers in Arms .ii	
Dungeon Hunter .iii	
Ice Age .iv	
Playdead	- 9
Limbo .i	
Inside .ii	

کنسول های بازی



PlayStation	- 1
Xbox	- 2
Nintendo	- 3

مشهورترین موتورهای بازی سازی

Unreal Engine	- 1
Amazon Lumberyard	- 2
CryENGINE	- 3
Unity	- 4
GameMaker: Studio	- 5

ژانرهای بازی

۱. مبارزه Fighting

بازی‌های مبارزه‌ای جزو اولین ژانر بازی‌های ویدئویی بودند و هنوز هم جایگاه ویژه‌ای دارند. در این بازی‌ها بازیکن می‌تواند یکی از شخصیت‌های متعدد را برای مبارزه انتخاب کند. هر کاراکتر ویژگی‌های خاص خودش را دارد؛ بعضی سرعت بیشتری دارند، بعضی قدرت بیشتری و بعضی هم در انجام حرکتی نسبت به دیگری قوی‌تر عمل می‌کنند.

مبارزات ژانر Fighting اغلب به صورت نبرد تن به تن هستند و بدون استفاده از سلاح اتفاق می‌افتند؛ مگر در مواردی که آن سلاح می‌تواند چاقو، شمشیر یا دیگر موارد سلاح سرد باشد. نسل بازی‌های مبارزه‌ای ابتدا با مورتال کمبت Mortal Combat در دوره سگا رو آمد و بعد با Tekken در دوران پلی استیشن جا افتاد. این روزها می‌توانید ورژن جدید این بازی‌های قدیمی را با کنسول‌های خانگی جدید مثل پلی استیشن و ایکس باکس بازی کنید.

۲. بقا Survival

یکی از غرایز اصلی انسان بقا است و نویسندگان بسیاری از بازی‌ها از این غریزه برای جلو بردن داستان‌شان استفاده می‌کنند. در بازی‌های ژانر Survival هدف نهایی بازیکن جان سالم به در بردن در شرایط سخت بازی است. برای مثال ممکن است فرد بازمانده سقوط یک هواپیما در یک جزیره غریبه باشد و قرار باشد بدون امکانات خاصی خودش را زنده نگه دارد. در این بازی‌ها به مرور و با کسب امتیاز بیشتر امکان تهیه اقلام ضروری و وسایل مورد نیاز برای زندگی به بازیکن داده می‌شود.

۳. Horror Survival

در بازی‌های ژانر وحشت و بقا اغلب محیطی فانتزی خلق شده است که بازیکن در آن با رویدادهای عجیب رو به رو می‌شود و باید بتواند خودش را نجات دهد. برای مثال بازی Resident Evil یکی از بازی‌های بسیار مشهور و موفق این ژانر است. ماجرای این بازی در شهری اتفاق می‌افتد که ساکنان آن زامبی هستند. کارکترهای اصلی یک پلیس جوان و دختری به نام کلیر هستند که باید خودشان را نجات دهند و به دنبال سرخ‌هایی باشند. این سبک برای افرادی که به تماشای فیلم‌های ترسناک علاقه دارند، ژانر جذابی خواهد بود.

۴. تیراندازی Shooter

سبک بازی‌های شوتر یا تیراندازانه یکی دیگر از ژانرهای محبوب و رایج بازی‌های ویدئویی است. در این بازی‌ها شما به عنوان بازیکن از طریق سلاح گرم به ویژه اسلحه روند بازی را جلو خواهید برد. بازی‌های شوتر معمولاً از دو زاویه دید طراحی می‌شوند؛ بعضی‌ها **زاویه دید اول شخص** دارند، یعنی دوربین روی شانه بازیکن وجود دارد و بازیکن خودش را نمی‌بیند؛ به همین دلیل هم‌ذات‌پنداری بالایی با شخصیت بازی‌اش خواهد داشت. در **زاویه دید سوم شخص** شما می‌توانید بازیکن را ببینید و او را هدایت کنید. میزان هم‌ذات‌پنداری در این زاویه دید کمتر از نوع اول شخص است. بازی *Half-Life*، *Doom* و *Call of Duty* از بهترین بازی‌های محبوب این ژانر هستند.

۵. پنهان کاری Stealth

بازی‌های ژانر پنهان کاری یا *Stealth* شامل آن دسته بازی‌هایی می‌شوند که بازیکن در آن باید ماموریتش را با مخفی کاری و ایجاد کمترین جلب توجه و سر و صدا انجام دهد. به همین دلیل هم این بازی‌ها هم به تمرکز بیشتری نیاز دارند و هم می‌توانند تمرکز را در افرادی که برای پرورش آن به تمرین نیاز دارند، بالا ببرند. اگر یکی از بازی‌های جدید سبک مخفی کاری را بازی کنید، احتمالاً می‌دانید که در بعضی شرایط ممکن است بازیکن گیر بیفتد و مجبور شود برای نجات خودش کار را به برخورد شدید و سر و صدا بکشانند. در این شرایط بازیکن مرحله را رد خواهد کرد؛ اما معمولاً یا امتیاز از او کم خواهد شد یا اینکه در رسیدن به مراحل بالاتر دچار دردسر خواهد شد.

۶. هک و اسلش hack and Slash

اگر دنبال کننده معروف‌ترین بازی‌ها باشید، احتمالاً اسم بازی *God Of War* به گوشتان خورده است. گاد آف وار یا خدای جنگ یکی از بازی‌های مطرح ژانر هک اند اسلش است. در این سبک بازی وظیفه بازیکن مبارزه با تعداد زیادی از دشمنان است که با ضرب آهنگی بسیار تند و فضا سازی خشن و سلاح‌های قدرتمند اتفاق می‌افتد. این بازی‌ها برای کسانی که به هیجان زیاد علاقه دارند و ضمناً با سر و صدا و شلوغ کاری زیاد هم مشکلی ندارند، جزو بازی‌های ایده‌آل محسوب می‌شوند. دشمنان شما در این بازی‌ها اغلب موجودات عجیب الخلقه با توانایی‌های منحصر به فرد هستند.

۷. نقش سازی Role playing

نقش آفرینی یا نقش سازی به اندازه ژانرهایی مثل شوتر و فایتر همه‌پسند نیست؛ اما مخاطبان خاصی را دارد که عاشق این سبک داستانی هستند. در سبک نقش آفرینی **بازیکن کنترل بسیار زیادی روی روند داستان و حتی اتفاقات داستانی دارد**. او می‌تواند شخصیت خودش را نقش آفرینی کند و او را وارد بازی کند. در قلمروی مخصوصش ماجراجویی کند یا با مشکلات و دشمنانی که سر راهش قرار می‌گیرند، مبارزه کند. تمام این کارها منجر به بالا رفتن مهارت‌های نقش بازیکن می‌شوند. بازی‌هایی مثل ویچر The Witcher (جادوگر) در دسته بازی‌های نقش آفرینانه قرار می‌گیرند.

نکته: ژانر Simulation یا شبیه سازی چیزی شبیه نقش آفرینی است؛ منتها در قالب آفرینش محیط جدید؛ برای مثال مزرعه یا کارخانه.

۸. معمایی Puzzle

کسانی که به چالش‌های فکری علاقمند هستند، احتمالاً ژانر معمایی را می‌پسندند. این بازی‌ها شامل چالش‌هایی برای فکر هستند و بازیکن باید با استفاده از نشانه‌ها، فکر کردن و پیدا کردن ارتباطات خودش را به مراحل بالاتر برساند. بر اساس علاقه‌تان می‌توانید سراغ معماهای پلیسی بروید یا از روایت‌های داستانی دراماتیک برای کشف و پیش برد مراحل استفاده کنید.

۹. استراتژیک Strategic

در ژانر استراتژیک شما مغز متفکر یک ارتش یا یک گروه هستید؛ به همین دلیل هم اغلب زاویه دید شما از بالا و روی محدوده وسیعی است که در آن نیروهایتان مشغول عملیات هستند. این بازی‌ها نیاز به فکر کردن و برنامه ریزی دارند و به همین دلیل هم به عنوان بازی‌های استراتژیک در نظر گرفته می‌شوند. از معروف‌ترین بازی‌های این سبک وارکرفت Warcraft است که طرفداران جدی زیادی دارد.

۱۰. پلتفرم Platform

کمتر گیمری است که سوپر ماریو را نشناسد؛ در این بازی و بازی‌های پلتفرمی مشابه آن، **وظیفه بازیکن بالا و پایین پریدن از موانع و رد کردن آن‌ها، گرفتن امتیازهای موجود سر راه** و استفاده از قابلیت‌هایی مثل غلت زدن و خزیدن و غیره است. این بازی‌ها ممکن است حالت دو بعدی داشته باشند و بازیکن صرفاً بتواند از طرف چپ صفحه به سمت طرف راست حرکت کند.

۱۱. ورزشی Sport

همان طور که از اسمش پیداست، ژانر ورزشی یا اسپرت شامل به چالش کشیدن توانایی‌های بازیکن در زمینه‌های مختلف ورزشی می‌شود. برای مثال بازی معروف **Need for Speed** یکی از بازی‌های قدیمی اما پرطرفداری است که شامل مسابقات اتومبیل رانی می‌شود. اما مشهورترین بازی‌های ژانر ورزشی اغلب شامل فوتبال می‌شود که شناخته‌ترین بازی آن بازی فیفاست.

ژانرهای موسیقی

امبینت (Ambient)

امبینت طیف گسترده‌ای از انواع موسیقی را در بر می‌گیرد و تمرکز آن، بیشتر از ساختار موسیقایی، بر روی صداهاست. هدف نهایی امبینت این است که با کمک قطعات و کال، سازهای متعارف و غیرمتعارف، آهنگی خلق کند تا حس و حال (Mood) خاصی را به مخاطب القاء نماید. این سبک می‌تواند دارای صدای خواننده هم باشد ولی بخش عمده آن با تجهیزات موسیقی بی کلام اجرا می‌شود. امبینت محدود به مود یا صدای خاصی نیست و از این ایده پیروی می‌کند که می‌توان بدون وابستگی به ساختارهای تئوریک موسیقی و خوانندگی، هر نوع جو و اتمسفری را ایجاد کرد. امبینت با توجه به گستره کاربردهایش در بسیاری از سبک‌های موسیقی به کار می‌رود.

بلوز (Blues)

موسیقی بلوز یک سبک آفریقایی-آمریکایی متعلق به مناطق جنوبی ایالات متحده بوده و از دل ترانه‌های کارگران، تصنیف‌ها و اشعار غمگین بیرون آمد و تا پایان قرن نوزدهم در همین سبک باقی ماند. این موسیقی در یک هارمونی ۱۲ میزانی نواخته می‌شد که البته محدود به این شکل باقی نماند و بعدها "نت‌های بلو" که تا حدودی با موسیقی وسترن کلاسیک در تضاد بود به آن اضافه شد.

در نیمه اول قرن بیستم، آمریکایی‌های آفریقایی تبار با ترکیب جاز گاسپل و بلوز، سبک جدیدی به نام ریتم اند بلو (Rhythm & Blue) به اختصار آر اند بی را ابداع کردند. این ژانر بعدها به زیرشاخه‌های دیگری مثل دلتا بلو، پیدمونت، جامپ بلوز و شیکاگو بلوز تقسیم شد. بعد از جنگ جهانی دوم بسیاری از نوازنده‌ها به خصوص در شهر شیکاگو، به الکترونیک بلوز گرایش یافتند که یک سبک موسیقی متأثر از ژانر راک اند رول بود. همین داستان برای سایر موزیسین‌ها در سبک بلوز راک، هارد راک و هوی متال هم افتاد. از آن زمان به بعد مخصوصاً در حوالی دهه ۲۰۰۰ با روی کار آمدن گاراژ راک، این سبک جان تازه‌ای گرفت.

موسیقی کلاسیک (Classical)

موسیقی کلاسیک غنای تاریخی و سیطره وسیعی بر نقاط مختلف دنیا داشته و به همین خاطر آن را می‌توان پایه و اساس ژانرهای موسیقی قرن بیستم دانست. تمامی سبک‌های موسیقی، خواه ناخواه ریشه در فرهنگ جوامع دارند ولی با این وجود باز هم ذیل موسیقی کلاسیک که مادر تمام ژانرهای موسیقی است قرار می‌گیرند. این سبک دوشادوش موسیقی فولکور یا سنتی، مسیر خود را طی می‌کند.

موسیقی کلاسیک ضمن حفظ مشخصه های فرهنگی، به لحاظ سبک، تنوع گسترده تری از فولکور دارد و همواره روی آن اثرگذار بوده است. در واقع این دو سبک موسیقی که به فرهنگ جامعه گره خورده اند نوعی همپوشانی با هم دارند. با این حال چند ویژگی مشترک وجود دارد که از طریقشان می توان فهمید کی و چگونه یک ژانر فولکور در دسته بندی کلاسیک قرار می گیرد. می توان گفت که موسیقی کلاسیک با توجه به جذابیت و ساختار پیچیده تر، بیشتر از فولکور مورد توجه قشر روحانیت، حاکمیت و علماست. مسلماً چنین حمایتی به موزیسین ها انگیزه می دهد تا با قدرت بیشتری موسیقی را گسترش دهند.

با توجه به فقدان موسیقی کلاسیک در جوامع غیر سلسله مراتبی به نظر می رسد که حمایت از چنین آهنگ هایی برای غنا بخشیدن به جنبه های اقتصادی، اجتماعی، علمی و مذهبی جامعه ضروری است. همچنین تمایز بین موسیقی کلاسیک و فولکور را می تواند منعکس کننده این تمایز پنداشت. این نوع موسیقی با نام های مختلفی در سراسر دنیا شناخته می شود از جمله «موسیقی جدی» در لهستان، «موسیقی مناسب» در کره و به صورت عمومی «موسیقی هنری».

کمدی (Comedy)

کمدی طیف وسیعی از سبک های موزیکال و غیر موزیکال را در بر می گیرد. ولی قاعده مشترک در ژانرهای منشعب از کمدی طنزمايگی آن است. کمدی موزیکال حقایق فرهنگی جامعه را به شکل طنز ارائه می دهد و معمولاً برای این کار از یک شعر خنده دار یا آواز رپ استفاده می کند. کمدی غیر موزیکال در بر گیرنده سبک های دیگری مثل اسکچ کمدی، استند آپ کمدی و شوخی تلفنی است.

کانتری (Country)

کانتری یک سبک پرطرفدار موسیقی است که چیزی در حدود یک قرن پشتهوانه تاریخی دارد. منشأ این سبک به ناحیه آپالاشیا شرق ایالات متحده و نواحی جنوبی آمریکا بر می گردد که آن زمان تحت سلطه موسیقی فولکور ایرلندی و انگلیسی بود. اولین خودنمایی گسترده سبک کانتری در سازهای کمانچه (fiddle)، گیتار، بانجو و سایر سازهای زهی در اوایل دهه ۱۹۲۰ ظاهر شد. در اواخر دهه ۱۹۲۰ و اوایل ۱۹۳۰، جیمی راجرز و خانواده کارتر جزو ستاره های این سبک بودند. استایل اجرای آن ها یودلینگ و کانتری گاسپل بود که به سبک اصلی کانتری در آن دوران بدل شد.

در سال ۱۹۳۰ و ۱۹۴۰ گرایش به سبک گاسپل و یودلینگ کاهش یافت و جای آن را سبک وسترن سوئینگ (Western Swing) و هانکی تونک (Honky Tonk) گرفت. شاخه دیگری از کانتری سبک بلوگرس بود که سال ها در شهرت باقی ماند. در این شرایط سبک های قدیمی موسیقی کانتری، موسیقی سنتی آمریکایی شناخته می شد که این موضوع اولین بار در نمودارهای جوک باکس در سال ۱۹۴۰ نمایان شد. اولین نمودارهای

جوک باکس درباره سبک کانتری تا سال ۱۹۴۹ تحت عناوین juke box و juke box hillbilly records و folk records و زیرگروه دو ژانر وسترن و کانتری عنوان می شد.

هانکی تونک تا ۱۹۵۰ در شهرت باقی ماند ولی از اوایل ۱۹۵۰ کارهای هنک ویلیامز بر زبان افتاد ولی در اواسط این دهه سبک کانتری تنها ژانر مورد علاقه مردم نبود. به عنوان مثال کارهای سبک Rockabilly نیز که حلقه ارتباطی موسیقی hillbilly و راک اند رول بود در محبوبیت قرار داشت. ولی بحث اصلی تاثیر موسیقی کانتری، کانتری بویگی (Country Boogie) و R&B روی به وجود آمدن سبک راک است. در اواخر دهه ۱۹۵۰ به خاطر روی کار آمدن تجهیزات و تولیدات موسیقی بهتر به خصوص رونق گرفتن کار ارکسترها و شکل گیری زیر شاخه جدیدی از ژانر کانتری تحت عنوان Nashville sound هانکی تونک عملاً از شهرت افتاد.

در اواخر دهه ۱۹۶۰ و در طول دهه ۱۹۷۰ سبک کانتری پاپ حد و مرز بین ژانر کانتری و سایر آهنگ های اصیل آمریکایی را از بین برد. همچنین سبک کانتری راک بیش از آنکه توجه مخاطبان کانتری را به خود جلب کند باعث موفقیت بیشتر گروه های موسیقی راک شد. در هر صورت این تغییرات نتوانست برای مدت طولانی طرفداران سبک کانتری را راضی نگه دارد .

موسیقی کانتری به رشد خود ادامه داد تا زمانی که محبوب ترین ژانر موسیقی ایالت متحده به خصوص در قسمت های جنوب و میانه غرب شد. گاهی برای ایجاد تمایز بین کانتری معاصر از آن چیزی که متعلق به ۳۰ سال قبل و هانکی تونک بود از عنوان «کانتری جدید» یا «کانتری اصلی» استفاده می کنند. ولی در میان سنت گرایان جریانی شکل گرفت که نتیجه آن بازگشت کانتری به همان عقبه تاریخی اش بود.

دنس (Dance)

دنس را می توان آغازگر سبک های جدید موسیقی مثل دیسکو، الکترونیک دنس و دنس پاپ دانست و همانطور که از نامش پیداست هدف آن خلق یک موسیقی شاد و مناسب رقص است.

الکترونیک (Electronic)

موسیقی الکترونیک گونه ای موسیقی است که سازهای الکترونیک ساختار اصلی آن را تشکیل می دهند. این موسیقی که اغلب بی کلام اجرا می شود در اوایل تحت سلطه افرادی مثل Edgard Varèse and Karlheinz Stockhausen بود و اغلب ترکیبی از کلاژهای مجازی حاوی لوپ ها، ساینر، اسیلاتورها و ... بود. اولین آهنگسازی که با ملودی و ریتم یک موسیقی الکترونیک درست کرد رایموند اسکات بود .

از سال ۱۹۶۰ به بعد با آغاز فعالیت گروه هایی مثل Silver Apples و Noise White موسیقی الکترونیک کم کم وارد سبک پاپ شد تا آنجا که نهایتاً منجر به ظهور سبک هایی مثل سینث پاپ (Synthpop) یا تکنوپاپ

شد. این موسیقی عامه پسند بوده و معمولاً هسته اصلی آن را سینتی سایزرها تشکیل می دهند. البته این سبک مخالفانی هم دارد که معتقدند الکترونیک یک موسیقی مرده و بی روح است.

اکسپریمنتال (Experimental)

این سبک که به موسیقی تجربی هم معروف است، آزادترین نوع آهنگ در تاریخ موسیقی است. آهنگ های اکسپریمنتال بیشتر در اواسط قرن بیستم میلادی و در آمریکای شمالی ظهور کردند. تولید موسیقی در این سبک بسیار منعطف است. هر نوع سازی مثل گیتار، ساکسوفون، ویولنسل و حتی صدای انسان در این آهنگ ها می تواند حضور داشته باشد. در این موسیقی آهنگساز می تواند از چیزهای بسیار غیر قابل انتظار استفاده کند که در زیرشاخه هایی مثل موسیقی اینداستریال و *Musique concrète* به وفور از آن استفاده می شود. در این روش از ترکیب چندین صدا یا افکت کلاژهای صوتی تولید می شود.

صدایی که به این روش تولید می شود معمولاً فراتر از مرزهای موسیقی سنتی است. نتیجه بهره گیری از سازهای سنتی خارج از ساختارهای آکادمیک، چیزی شبیه بداهه نوازی یا امپروایزیشن است که مسلماً برنامه ریزی خاصی ندارد. صرف نظر از نوع اجرا، آهنگ های اکسپریمنتال برخلاف سایر سبک های موسیقی، ریتم و تمپوی مشخصی ندارند و بافت آهنگ نیز می تواند تخریب شده باشد.

ریشه های موسیقی اکسپریمنتال را می توان در موسیقی کلاسیک (قرن ۲۰ ام میلادی) در جنبش هایی همچون فوتوریسم یافت. اولین موزیسین هایی که به صورت آکادمیک وارد این سبک شدند افرادی چون *John Cage* ، *Pierre Schaeffer* و *Luigi Russolo* بودند ولی هنرمندانی خارج از دانشگاه همچنان به روش بداهه نوازی و جز آزاد در این سبک فعالیت می کردند که نوعی ادغام تجربیات موسیقایی در آهنگ بود. با همین روند افراد دیگری نیز همچون *Einstürzende Neubauten* و *Supersilent* به روش تجربی رو آوردند. از اواخر قرن بیستم با محبوبیت سازهای الکترونیک مثل سینی سایزر و نرم افزارهای آهنگسازی افراد بیشتری وارد حوزه آهنگسازی شدند یا حداقل توانستند تجربه آهنگسازی داشته باشند.

گفتنی است که امروزه واژه اکسپریمنتال یا تجربی به عنوان پسوند در ادامه بعضی از سبک های موسیقی قرار می گیرد که معنی اش این است که آن سبک با تغییراتی در ساختار یا ترکیب اصلی همراه است. این تغییرات همانطور که از نام آن ها پیداست تجربی و برای ایجاد نوعی غیر معمول یا نوینی از آن سبک به کار می رود. این لفظ حتی به موسیقی الکترونیک هم نسبت داده می شود. به طور خلاصه اکسپریمنتال به نوعی موسیقی گفته می شود که سبک خاصی نداشته و در جریان کلی موسیقی جای مشخصی ندارد.

فیلد رکوردینگ (Field Recording)

فیلد رکوردینگ به خاطر شباهت زیاد به فونوگرافی یا صدانگاری گاهی به همین نام نیز نسبت داده می شود. **فیلد رکوردینگ به هر متریال ضبط شده خارج از استودیو اطلاق می شود؛** از کپچر کردن صدای طبیعت مثل صدای آب، باد، حیوانات تا صدا از زندگی روزمره مثل قدم زدن و همه‌مردم، صدای بازی کردن بچه ها، حرکت قطار، ترافیک و شلوغی شهر و ... این صداها معمولاً در ژانرهای دیگر موسیقی مثل اکسپریمنتال، امینت، *Musique concrète* و ... به کار می روند. این سبک را می توان در تعریفی عام به هر چیزی که خارج از استودیو ضبط می شود نسبت داد اعم از مکالمه، موسیقی زنده و ...؛ ولی چیزی که بیشتر به این ژانر مربوط می شود ضبط صدای محیط است.

فولک (Folk)

فولک یک اصطلاح بسیار کلی برای موسیقی به شمار می رود که ریشه در سنت های شفاهی جوامع بشری دارد. این موسیقی در شمار موسیقی سنتی یا به طور دقیق تر، موسیقی فولکور قرار می گیرد که در اوایل قرن بیستم رشد کرد و بعدها در دهه ۶۰ به اوج محبوبیت خود رسید. این موسیقی غالباً از دل توده های مردمی همچون کارگران، اجتماعات مذهبی و خانواده ها سر درآورده است. در این تعریف مردم به هر گروهی اعم از ملیت ها، قبایل، اقوام، مشاغل، خانواده ها و ... یا به موسیقی سنتی مناطق مختلف همچون موسیقی بومیان آمریکا، موسیقی قبیله هیل تریب، موسیقی فولکور کانادایی، ترانه های کار و موسیقی فولک شرق آسیا اطلاق می شود.

در حال حاضر موسیقی فولک به دو دسته کلی تقسیم می شود: فولک سنتی و فولک معاصر. فولک سنتی که زیردسته موسیقی محلی محسوب می شود انواع مختلفی از آهنگ های محلی را شامل می شود. این سبک را می توان زیربنای اصلی موسیقی فولک دانست. با نگاهی به تاریخچه فولک می توان ردپایی از دستکاری ترانه ها و تکامل آهنگ ها را دید که در قالب ریتم هایی مشابه اما در مقیاس های مختلف نواخته شده اند. در اوایل تلاش می شد این سبک را به دو دسته روستایی و شهری تقسیم کنند ولی امروز همانطور که می بینیم این رویکرد به صورت گسترده ای رد شده است. در اواسط قرن بیستم، موسیقی فولک معاصر به دسته ای از آهنگ ها اطلاق می شد که ریشه هایی از موسیقی فولک اروپایی و آمریکایی در آن وجود داشت.

هیپ هاپ (Hip Hop)

هیپ هاپ به طور عمده در سواحل شرقی ایالات متحده در اواخر ۱۹۷۰ روی کار آمد. این موسیقی به جای استفاده از ملودی های هارمونیک یا خوانندگی و کال، بر ریتم آهنگ و الگوهای آوایی تأکید دارد. ریشه های اصلی این سبک را می توان در سبک دی جی ردیابی کرد که خود نوعی موسیقی رگه

(Reggae) به حساب می آید و در آثار DJ Kool Herc می توان نمونه های آن را دید. مشتقات دیگر هیپ هاپ مثل دیسکو و فانک باعث شد تا این سبک از کوچه و خیابان وارد اجتماعات شهری شود طوری که در فاصله سال ۱۹۸۰ تا ۱۹۹۰ شهر نیویورک و سپس کل آمریکا و همچنین سایر کشورها با هیپ هاپ آشنا شدند. نهایتاً با دخالت دو ژانر جز و سول، هیپ هاپ به شکل امروزی خود درآمد.

جز (Jazz)

جز در جوامع آفریقایی-آمریکایی در جنوب ایالات متحده حوالی قرن بیستم ریشه دوانده است. جز متأثر از دیگر سبک های موسیقی همچون بلوز و رگتایم و همچنین تیم های اثرگذار موسیقی همچون نیوئورلئان براس باند که خود آمیخته ای از موسیقی شرق آفریقا و موسیقی ارتش اروپا بود در دهه ۱۹۳۰ با ابداع زیرشاخه ای از جز به نام سوئینگ خیلی سریع خودش را به عنوان یک ژانر اثرگذار موسیقی به ثبت رساند. صداهای به کار رفته در جز در قرن اخیر دستخوش تغییرات زیادی شده است ولی کماکان بداهه نوازی المان اصلی آهنگ های جز را تشکیل می دهد.

ژانر بلوز در اوایل ساختاری متشکل از الگوی تکراری ندا-پاسخ بود که شکلی از موسیقی فولک در میان سیاه پوستان آمریکایی واقع در مناطق روستایی محسوب می شد و نوعی بداهه نوازی به حساب می آمد. در کنار بداهه نوازی، نت های موسیقی بلوز و پلی ریتم های همزمان به کار رفته در آن شاخص خوبی برای شناخت پایه های اصلی سبک جز است. این ژانر به شکل موثری واژگان موسیقایی خود را وارد تئوری موسیقی کرده تا ویژگی های منحصر به فردی همچون سوئینگ را که نوعی احساس ریتمیک با فواصل مشخص در بین پر فرمها است و استفاده وسیع از ii-V-I progression را بیان کند.

در میان اولین سبک های شناخته شده از این ژانر می توان به دیکسی لند متعلق به نیوئورلئان در سال ۱۹۱۰ و استراید که شکلی از موسیقی پیانو و توسعه یافته سبک رگتایم است اشاره کرد. جز تا دو دهه به گسترش خود ادامه داد تا اینکه بعد از اولین محبوبیت بین المللی، بخش عمده تمرکز از سوئینگ که مبتنی بر رقص بود به سبک ریتمیک پیچیده تری به اسم بی پاپ در اواسط دهه ۴۰ معطوف شد.

حرکت به سمت بی پاپ باعث شد تا بر ویژگی های موجود در تک نوازی های تک نفره که مشخه جز در دهه های بعدی بود تمرکز بیشتری شود. بعدها در مقابل این سبک پر شور نوعی از جز تحت عنوان کول جز هم که نسخه آرام تر با تکنیک نوازندگی ملایم بود روی کار آمد. ارنجمنت کنترل شده در این سبک اجازه می داد تا به تقلید از موسیقی کلاسیک جریان سومی در موسیقی جز شکل گیرد. بعدها جز آفریقایی-کوبایی و سامبا-جز که الهام گرفته از موسیقی برزیل بود نهایتاً جز را به شکل غربی امروزی نزدیک کرد.

ساخت زیر شاخه های جدید جز ادامه یافت و سبک مدال جز که به جای آکوردها از حالت موزیکال در چارچوب اصلی استفاده می کرد به وجود آمد. ده سال بعد فری جز ظهور کرد که خود زمینه ای برای تولد سبک دیگر آوانت گرید جز بود. اواسط دهه ۶۰ در بحبوحه جنبش حقوق مدنی آمریکا، جان کلترین زمینه ساز ظهور سبک جدیدی از جز به نام جز معنوی یا اسپیرچوال شد که متأثر از طیف متنوعی از آهنگ های مذهبی، غیر وسترن و به ویژه موسیقی رایج در جنوب شرق آسیا بود.

تداوم ادغام سبک های موسیقی و خلق زیرشاخه های جدید جز به خصوص ژانرهای موسیقی آفریقایی-آمریکایی سبک های دیگری از جز مثل جز فانک را روی کار آورد که شباهت های زیادی با سول جز داشت. با افزایش محبوبیت موسیقی راک در دهه ۷۰ جز وارد راک هم شد طوری که سبک های دیگر جز مثل جز راک و جز فیوژن متولد شدند که روح بداهه نوازی هارمونیک در جز را با تجهیزات نوازندگی الکترونیک ترکیب می کرد و خروجی آن یک موسیقی ریتمیک راک بود.

البته این ترکیب خالی از انتقاد نبود و بسیاری از دوست داران جز، سبک فیوژن را اصلاً ایده خوبی نمی دانستند چون معتقد بودند چنین ترکیبی با روح جز مخالف است. افزون بر این بسیاری از مبدعان سبک جز از جمله مایلز دیویس، از سبک های هارد باپ به فیوژن رفتند و از آنجا به جز ملایم یا نرم رسیدند. در دهه ۱۹۹۰ با محبوبیت موسیقی الکترونیک، جز وارد عصر جدیدی به نام نئو جز شد که تجربه جدیدی از جز به شنوندگان خود منتقل می کرد.

متال (Metal)

با اینکه ریشه دقیق هوی متال مشخص نیست ولی می توان برآورد کرد که این ژانر در اواخر دهه ۱۹۶۰ توسعه یافته است. هوی متال تأثیر زیادی از گروه های افسانه ای بلوز راک همچون، Led Zeppelin Cream, The Rolling Stones و The Who گرفته است.

صرف نظر از اینکه آغازگر هوی متال چه گروهی یا چه کسی بوده، این ژانر در اوایل ۱۹۷۰ به محبوبیت زیادی دست یافت و در انگلستان و ایالات متحده گروه های بزرگی در سبک هوی متال روی کار آمدند. گروه هایی مثل Motörhead and Thin Lizzy. Judas Priest ، AC / DC

خصیصه اصلی متال در اوایل، شکل گیری خشونت بود؛ خشونت در نواختن ریف ها، درام های خشن، تک نوازی های طولانی (اغلب گیتار) و آواز خوانی با صدای بلند. در واقع در این سبک موسیقی حس شلوغی و خشونت بیش از هر چیز دیگری شنیده می شود. نوع پوشش نیز در اجرای صحنه ای متال

متمایز است؛ غالب رنگ های به کار رفته در صحنه مشکی است و معمولاً پرفرمرها لباس تنگ چرمی به تن دارند .

اغلب خواننده ها و نوازنده های این ژانر به تبعیت از جنبش ضدفرهنگ هیپی در ۱۹۶۰، موهای مشکی بلند دارند . شعرهای به کار رفته در متال غالباً ضد ارزش های فرهنگی است و جنبه های تاریک و افسرده ای مثل شیطان پرستی، بی خدایی، مواد مخدر، جنگ، مرگ و خودکشی را هدف قرار می دهد.

نماد اصلی هوی متال فرشته اخراج شده از بارگاه الهی یعنی ابلیس است و چیزی که می توان به صورت پررنگ در این سبک دید ترویج شیطان پرستی، رفتارهای ضد اجتماعی و توصیه به فردیت و تنهایی است.

از اوایل ۱۹۷۰، متال به زیرشاخه های زیادی تقسیم شد که هر کدام تم و ویژگی خاصی داشتند. به عنوان مثال (Death metal) زیرشاخه ای بسیار تهاجمی از متال محسوب می شود که موسیقی آن گوش خراش همراه با درام ها و تمپوی سریع با ساختار موسیقایی پیچیده و تأکید بر جنایت و کارهای خلاف عرف است .

از سوی دیگر بلک متال زیرشاخه ای از متال است که بیشتر به تولید موسیقی لو-فای (Lo-Fi) تمرکز دارد تا شدت اثرگذاری اصوات خشن، ترانه های غیرمعارف و اشعار شیطانی و ضد مسیحیت را بیشتر کند. با اینکه سبک های مختلف متال تفاوت های زیادی با هم دارند ولی در همه آن ها پیام اصلی متال که از اوایل شکل گیری وجود داشت حضور دارد یعنی: فرد گرایی.

عصر جدید (New Age)

موسیقی عصر جدید اغلب روی ملودی های مسالمت آمیز و آرام بخش (برای مدیتیشن) تمرکز دارد. این موسیقی انواع مختلفی دارد: مثلاً عصر جدید الکترونیک مبتنی بر سینتی سایزر، نت های کشیده و بلند و ضرب آهنگ های ساده درامز است. عصر جدید آکوستیک از سازهایی مثل فلوت، پیانو و گیتار برای ایجاد فضایی آرام استفاده می کند که گاهی از نمونه های دیجیتال این سازها نیز استفاده می شود.

از معروف ترین آرتیست های این سبک می توان به Mike Oldfield, Enigma, Popol Vuh و Vangelis اشاره کرد. تعدادی از هنرمندان این سبک نوعی موسیقی فولک یا سنتی را هم وارد کار خود کرده اند تا صدایی خاص و متمایز ایجاد کنند. به عنوان مثال Enya و Clannad افرادی هستند که عصر جدید را با موسیقی عامیانه سلنیک ترکیب کردند تا ژانر عصر جدید سلنیک را پدید آورند. به همین ترتیب CUSCO و Medwyn Goodall که موسیقی آمریکایی را به سبک Andean New Age اجرا می کردند.

پاپ (Pop)

موسیقی پاپ بیش از سایر ژانرها تحت تأثیر اقتصاد، تولید و بازاریابی انبوه قرار دارد که اولین تلاش برای همه گیر شدن آن در قرن نوزدهم صورت گرفت. این نخستین بار بود که ترانه سرایان می توانستند بدون سفرهای تبلیغاتی یا حمایت از گروه های موسیقی، درآمد مستقل داشته باشند. اختراع سیلندرهای فونوگرافیک در پایان قرن نوزدهم و مهم تر از آن تکنولوژی دیسک های موسیقی در اوایل دهه بیستم، استاندارد آهنگسازی و پتانسیل نشر انبوه آثار موسیقی را یک گام بلند، به جلو راند.

آهنگ های غیر کلاسیک در نیمه نخست قرن بیستم عمدتاً موسیقی محبوب (Popular) شناخته می شدند و نه پاپ. با این حال، موسیقی محبوب، از ۱۹۰۰ تا ۱۹۴۰ که مطابق میل مردم آمریکا است غالباً به عنوان موسیقی پاپ سنتی شناخته می شدند. اقتصاد رو به رشد بعد از جنگ جهانی دوم و ظهور دوران راک اند رول دوران جدیدی از مصرف گرایی را بین نوجوانان رقم زد و در همین دوران نیز موج اول سلبریتی های پاپ سر از لاک در آوردند. در بسیاری از منابع، توصیه شده واژه پاپ را برای قبل از ۱۹۵۰ اعمال کنیم ولی سال ۱۹۵۰ بود که پاپ به عنوان یک پدیده همه گیر فرهنگی پدیدار شد.

از آنجایی که **Pop خلاصه شده Popular است و در برهه های زمانی مختلف موسیقی محبوب تغییر می کند، اطلاق این واژه به سبک مجزایی از موسیقی، کمی ناخوشایند است.** به هر حال، اغلب آهنگ های پاپ ویژگی های مشترکی چون جذابیت، ملودی جوان پسند، ترانه های امروزی و دسترسی آسان دارند که مطابق سلیقه عموم افراد، به ویژه جوانان است. ترانه های پاپ معمولاً توسط تیم های بزرگ موسیقی و آهنگسازان حرفه ای ساخته می شوند.

موسیقی پاپ بیشتر از موسیقی راک روی و کالیست یا خواننده تأکید دارد و ساختار موسیقی آن دارای نظم خاصی است. مثلاً در ابتدای همه آهنگ های پاپ یک مقدمه یا اینترو وجود دارد و در ادامه قطعات صدا با نظم خاصی قرار می گیرند. در دهه ۱۹۶۰ گروه انگلیسی بیتلز که جزو بزرگان سبک راک بودند با تغییر نت ها چالش جدیدی برای سبک پاپ ایجاد کردند. در دهه ۱۹۷۰ بسیاری از سافت راک ها مسیر خود را به موسیقی پاپ تغییر دادند. با پایان این دهه، سبک دیسکو با الگو گرفتن از پاپ، ژانر **dance-pop** را به وجود آورد که تمرکز آن روی رقص و حرکات موزون بود.

پاپ و R&B همواره در تقابل مستقیم و غیر مستقیم هم بوده اند. در دهه ۱۹۸۰ مرزهای موسیقی پاپ با روی کار آمدن اسطوره هایی مثل مایکل جکسون باز هم دستخوش تغییر شد. در قرن ۲۱ موسیقی هیپ هاپ بیشترین تاثیر را بر موسیقی پاپ گذاشت طوری که بسیاری از ستاره های پاپ، از سازندگان موسیقی هیپ هاپ برای آهنگ

های خود استفاده می کردند. سرانجام موسیقی پاپ محبوبیت جهانی یافت (پاپ ایرانی، پاپ هندی، پاپ کره ای و ...). و این موضوع به خوبی نفوذ این ژانر به عنوان یک جریان اقتصادی در بین مصرف کننده ها را نشان داد.

پانک (Punk)

پانک یک از انواع سبک های موسیقی است که مشهور به سادگی، صداقت و مردمی بودن است. ریشه های پانک را می توان به Proto-Punk و گاراژ راک به سال ۱۹۶۰ نسبت داد ولی موفقیت اصلی این ژانر یک دهه بعد در سبک پانک راک پدیدار شد. از آن زمان پانک در سایر گرایش ها مثلاً پانک هارد کر و سایر استایل ها مثل الکترونیک، متال و اسکا توسعه یافت.

آر اند بی (R&B)

سبک R&B (شکل اختصاری Rhythm & Blues) از موسیقی آفریقایی-آمریکایی که در اواخر دهه ۱۹۴۰ نخستین بار واژه ریتم اند بلوز مطرح شد تا شکل معاصر R&B در دهه ۱۹۸۰ را در بر می گیرد. در خلال زمانی که عنوان R&B ناپدید شد (یعنی اواخر دهه ۱۹۶۰) تا ظهور مجددش در ۱۹۸۰ دو ژانر سول و فانک در اجتماعات آفریقایی-آمریکایی محبوبیت زیادی یافتند. این دو ژانر به شدت تحت تأثیر سبک R&B در دهه اول فعالیتش قرار داشتند به همین خاطر زیرمجموعه ای از آن محسوب می شوند. از ستاره های این ژانر می توان به افرادی چون Stevie Wonder، James Brown، Aretha Franklin، The Temptations اشاره کرد.

در اوایل دهه ۱۹۵۰، واژه فوق سبک های دیگری چون جامب بلوز، اشکال اولیه راک اند رول را هم در بر می گرفت. ولی چیزی که امروز بیش از بقیه آر اند بی خواننده می شود سبک هایی چون شکل اولیه ریتم اند بلوز، doo-wop آر اند بی نیوئورلئان است. با روی کار آمدن سبک هیپ هاپ در دهه ۱۹۹۰، چتر گسترده ای که واژه R&B روی انواع آهنگ های آفریقایی-آمریکایی افکنده بود به زمین افتاد. گرچه این ژانر در کنار هیپ هاپ به گسترش خود ادامه داد ولی باز هم گهگاهی این تاثیر در قالب ژانرهای ترکیبی جدیدی مثل New Jack Swing نمایان می شد.

ریتم اند بلوز، سول و فانک ژانرهایی بودند که تحت تاثیر موسیقی آمریکایی به ایجاد سبک های دیگری مثل بریتیش ریتم اند بلوز، لاتین سول و آفریبت کمک کرد. آر اند بی همچنین تاثیری زیادی روی سبک پاپ و دیسکو گذاشت. گفتنی است که واژه ریتم اند بلوز و مخفف آن R&B هر دو برای اولین بار در سال های پایانی دهه ۱۹۴۰ توسط بازاریابان برای اشاره به موسیقی سیاهان آمریکایی باب کرد.

این اصطلاح به عنوان جایگزینی برای *race music* استفاده شد که تا آن زمان برای اشاره به سبک موسیقی عامه پسند سیاهان آمریکا رواج داشت. ولی بعدها حالت مخفف، غالباً برای شکل معاصر *R&B* به کار رفت. بعلاوه این گفته که آر اند بی شامل سول و فانک می شود، دیگر به طور گسترده مورد قبول نیست و آن را کاملاً سبک مجزایی می دانند. آنچه مسلم است تاریخچه نمودارهای مجله بیلپورد است که سردرگمی زیادی برای مخاطبان به وجود آورد. به هر حال، امروزه *R&B* به شکل معاصر آن گفته می شود و کمتر به نسخه های اولیه در دهه ابتدایی آن اشاره دارد.

موسیقی منطقه ای (Regional Music)

موسیقی منطقه ای به سبکی از موسیقی اشاره دارد که سنت های فرهنگی خاص یک منطقه را منعکس می کند. ژانرهای موسیقی، زمانی منطقه ای بودن خود را از دست می دهند تا به چارچوب های فرهنگی و سلیقه افراد آن ناحیه مقید نباشند. هر موسیقی محلی بنا به نوع مخاطبان، متعلق به منطقه خاصی است. به همین دلیل موسیقی منطقه ای، موسیقی فولکور یا سنتی یا هر موسیقی دیگری که غنای فرهنگی زیادی در آن نهفته را شامل می شود.

سیر موسیقی منطقه ای از موسیقی ابتدایی و تاریخی آغاز می شود و راه خود را از میان ساختارهای پیچیده تری چون موسیقی فولکور، کلاسیک منطقه ای (کلاسیک عربی، کلاسیک هندوستان، کلاسیک ژاپنی و ...) به سبک های مدرنی که اساس آن ها موسیقی سنتی است مثل کیپ وردی، کالیپسوی کارائیب، ام پی بی برزیل، رومبا کانالانای اسپانیایی و ... باز کرده است.

موسیقی جهانی محدود به یک منطقه خاص نیست. با این حال سبک جهانی در شکل ابتدایی خود تکامل یافته موسیقی منطقه ای به ویژه موسیقی آمریکایی و اروپایی است که مثال مشهورش آهنگ های کلاسیک وسترن، بلوز و جز است. با اینکه این سبک ها ریشه های تاریخی در موسیقی سنتی دارند ولی به عنوان یک مفهوم جهانی ادراک می شوند و از حالت منطقه ای خارج شده اند.

از آنجایی که جهانی شدن به شدت تحت تأثیر فناوری است، پیشرفت های تکنولوژیک نقش مهمی در تمایز موسیقی منطقه ای و جهانی ایفا می کنند. موسیقی منطقه ای معمولاً از سازهای سنتی مانند بانجو، بالالایکا، سی تار و آکاردئون استفاده می کند؛ در حالی که موسیقی جهانی به طور گسترده ای از ابزارهای مدرن و الکترونیک همچون بیس، گیتار الکتریک، درام ماشین، سینتی سایزر و بسیاری از تجهیزات الکترونیکی دیگر استفاده می کند. بعضی از سازها به خصوص آن هایی که ارتباط نزدیکی با موسیقی وسترن کلاسیک دارند (پیانو، ویولون، کلارینت و ...) در هر دو سبک منطقه ای و جهانی و ژانرهای زیرمجموعه کاربرد دارند.

جدا از بحث ابزار، موسیقی منطقه ای تاکید ویژه ای روی تکنیک های سنتی آواز مثل یودلینگ، آواز خواندن از گلو، جوبک یا آواز پلی فونیک دارد. به علاوه، در طول سال های متمادی، در مناطق متعدد ریتم های جدید، مقیاس های زمانی و ساختارهای متنوع موسیقی به وجود آمده اند. به عنوان مثال Shomyo, Gagaku و تعدادی دیگر از سبک های متداول در موسیقی ژاپن از گام YO استفاده می کنند. گام های منحصر به فرد دیگری نیز در موسیقی ژاپن وجود دارند. این ویژگی ها در کنار ابزار و سبک آواز نمایانگر این است که ژانرهای منطقه ای چیزی بیشتر از صرفاً یک دسته بندی ساده جغرافیایی است.

گرچه موسیقی منطقه ای خلاف موسیقی جهانی است اما حجم قابل توجهی از انواع موسیقی وجود دارد که ترکیبی از هر دوی آنهاست. زیرشاخه های محلی ژانرهای جهانی در صورتی که از المان های منطقه ای مثل آفرو جاز، فانک لاتین، پاپ عربی و نظایر آن استفاده کنند جزو همین دسته محسوب می شوند. با این حال، ژانرهایی مثل پیکو پاپ، راک Yacht و polish yass صرف نظر از فرم منطقه ای موسیقی منطقه ای به شمار نمی روند چون در صدد گسترش موسیق جهانی هستند و نه منطقه ای.

راک (Rock)

راک یکی از محبوب ترین و پرکاربردترین ژانرهای موسیقی است که زیرشاخه های متعددی از آن پدید آمده است. ریشه های راک را می توان در آهنگ های اواخر دهه ۱۹۴۰ که حاوی المان های جامپ بلوز و سوئینگ بود و تا حدودی نقش گیتار الکتریک که از سبک شیکاگو بلوز گرفته شده بود در آن وجود داشت جستجو کرد. با اینکه اصطلاح «راک» که در ابتدا توسط هنرمندان بلوز در اواخر دهه ۱۹۳۰ برای توصیف نسخه پرتعداد آهنگ های خود که تمپوی سریع تری داشت به کار رفت تا اواخر دهه ۱۹۴۰ به طور گسترده به کار نرفت.

راک در شکل اصلی خود نوعی موسیقی مبتنی بر آواز بود که معمولاً از یک ساختار ورس (Verse) کورس (Chorus) با یک ریتم بک بیت و به کارگیری گیتار الکتریک به شکلی سریع تر و خشن تر از ژانرهای قبلی تشکیل شده است. ساختار راک و صدای آن مبتنی بر گیتار الکتریک و گیتار بیس است. ریتم آهنگ غالباً توسط درامز تولید می شود ولی گاهی خود گیتار هم برای این کار مورد استفاده قرار می گیرد.

با این حال، موزیسین های راک این ساختار بیس را در تمام آهنگ های خود تجربه کرده اند؛ بنابراین باندهای مختلف موسیقی راک اشتراکات زیادی با هم دارند. موسیقی راک از ابتدا جنبه های ضد فرهنگی را با هدف خدمت به شورش ها و اعتراضات هدف قرار داده که البته با ترویج راک، اهمیت این ویژگی کمرنگ تر شده است.

راک در جنوب ایالات متحده به وجود آمد و در دهه ۱۹۵۰ با دو شکل اصلی R&B و Rockabilly به محبوبیت رسید. راک اند رول برای توصیف آهنگ های ساخته شده توسط موزیسین های سیاه پوست همچون لیتل ریچارد و چانک بری به کار می رفت که با تمپوی سریعی R&B را اجرا می کردند و برای این کار از گیتار الکتریک، آمپلی فایر و میکروفون بهره می بردند. Rockabilly برای توصیف موزیسین های جوان وسترن سوئینگ به کار می رفت که R&B را در شکل سرعتی متأثر از موسیقی کانتری اجرا می کردند.

این اصطلاح بیشتر به موزیسین های سفید پوستی مثل کارل پرکینز اطلاق می شود و با آهنگ های الویس پرسلی به محبوبیت رسید. محبوبیت راک در اواخر دهه ۱۹۵۰ و اوایل دهه ۱۹۶۰ کاهش یافت چون در این مقطع، موسیقی پاپ طرفداران بیشتری پیدا کرده بود. ولی توسعه زیرسبک های راک مثل گاراژ راک و سارف راک تا حدودی محبوبیت از دست رفته را جبران کرد و این ژانر را از یادها نبرد.

موسیقی راک در انگلستان از اسکافل که یک ژانر مشتق یافته از جز و فولک است به وجود آمد. در دهه ۱۹۶۰ به راه افتادن رویدادی فرهنگی تحت عنوان British Invasion از سوی باندهای موسیقی Rolling Stones و Beatles کمک کرد تا مجددا توجه عمومی به این ژانر برگردد. این گروه ها تمایل داشتند عناصر بلوز و راک آمریکایی را بازسازی کنند و آنها را با موسیقی محبوب انگلستان مثل موسیقی بیت و مود که موجب تاثیر متقابل بین ایالات متحده و انگلیس می شد ادغام کنند.

در طول دهه ۱۹۶۰، موسیقی راک، پا را از ریشه های سنتی خود فراتر نهاد و استفاده از جنبه های غیرمتعارف گیتار مانند افکت ها و دیستورشن را، تا آنجا ادامه داد که راک از سبک ابتدایی خود فاصله زیادی گرفت. با تجاری شدن راک، سبک پاپ راک به عنوان نسخه ای محبوب و قابل دسترس در رادیو استفاده شد. همچنین در دهه ۱۹۶۰ پیشرفت های زیادی در فولک راک به وجود آمد که به طور گسترده ای در اعتراضات سراسر جهان به کار رفت و به دنبال آن ساکالوژیک راک که تحت تاثیر داروهای روان گردان تحت عنوان هلوسینوزنیک قرار داشت به وجود آمد. علاوه بر این در دهه ۱۹۶۰ راک پروگرسو شکل گرفت و استفاده از تجهیزات دیگر موسیقی مثل کیبورد و همچنین ترانه های پیچیده تر رواج یافت.

در دهه ۱۹۷۰، هارد راک، گاراژ راک و بلوز راک با خوانندگی تهاجمی تر و دیستورشن بیشتر گیتار مورد توجه عموم قرار گرفت. هارد راک اساس سبک متال را پایه ریزی کرد که بعدها به زیرشاخه های زیادی تقسیم شد. راک ثناتری و کمدی در اوایل دهه ۱۹۷۰ توسعه یافت و در نیمه اول دهه ۱۹۹۰ محبوبیت زیادی به دست آورد.

پانک راک در اواسط دهه ۱۹۷۰ تمرکز خود را بر روی صدای تهاجمی، ساختار موسیقایی پیچیده و همچنین جنبه های ضد فرهنگی گذاشت. پانک هر دو جریان قبل و بعد از دهه ۱۹۸۰ را تحت تاثیر قرار داد و زمینه را برای محبوبیت ژانرهای دیگر راک مثل موج نو، پست پانک و نهایتاً **Alternative Rock** و ایندای راک که تا چند دهه بعد در شهرت باقی ماند فراهم کرد. راک چندین دهه بعد به گسترش خود ادامه داد و به تعداد ژانرهای زیرمجموعه آن افزوده شد.

اسکا (Ska)

اسکا نوعی موسیقی است که در اواخر دهه ۱۹۵۰ در جامائیکا از دو سبک جز و ریتم اند بلوز (به طور خاص R&B نیواورلئان و جامپ بلوز) و ترکیبی از سبک های محلی منتو و کالیپسو به وجود آمد. نسخه اصلی اسکا به عنوان یک سبک موسیقی متمرکز بر ریتم های آف بیت شناخته می شود که از **horn section** ها، **busy bass lines** و هارمونی نرم و کال بهره می گیرد .

در اواسط دهه ۱۹۶۰ اسکا در ترکیب با سبک های دیگری مثل سول ژانر **Rocksteady** را به وجود آورد. گسترش این ژانر در کنار توسعه خود اسکا باعث به وجود آمدن ژانر دیگری به نام رگه در اواخر دهه ۱۹۶۰ شد. در دهه ۱۹۷۰ موزیسین های بریتانیایی که سال ها در گسترش اسکا فعالیت کرده بودند نوعی دیگری از این ژانر تحت عنوان **Tone 2** پدید آوردند .

به دنبال آن در اواسط تا اواخر دهه ۱۹۸۰ باندهای موسیقی در شمال آمریکا دوباره این سبک را -متاثر از ۲ - **tone** با هدف ساخت موج سوم اسکا جان دوباره دادند. ماحصل تمام این فراز و فرودها این شد که اسکا به یک پدیده جهانی بدل شد و سبک های مختلفی از دل آن بیرون آمد. تمام ژانرهای اسکا مبتنی بر آف بیت ها هستند و تمایل زیادی به استفاده از گیتار در فرم **chop-style** دارند. اسکا با تمپوی مناسبی برای رقص نواخته می شود.

صداها و افکت ها (Sound and Effects)

آثار صوتی غیر موزیکال در دسته صداها و افکت ها قرار می گیرند که شامل انواع مختلف اصوات تولید شده توسط سینتی سایزر و سایر تجهیزات الکترونیکی است. این صداها معمولاً به شکل انتزاعی هستند و وجود خارجی ندارند. این صداها و افکت ها در سایر سبک های موسیقی به خصوص موسیقی الکترونیک و سبک هایی مثل امبینت کاربرد زیادی دارند. البته در این بین ممکن است قطعات آهنگینی مثل بخش کوتاهی از یک آهنگ که موقع رد کردن موج های رادیو به گوش می رسد وجود داشته باشد ولی تمرکز اصلی این صداها به جای موسیقی روی جو صوتی ایجاد شده معطوف است. این صداها با صداهایی فیلد رکوردینگ که به صورت واقعی از طبیعت ضبط می شوند متفاوت است و همانطور که گفتیم غیرواقعی و انتزاعی هستند.

Spoken Word

همانطور که از نامش پیداست به آوای ضبط شده توسط انسان اشاره دارد. این دسته بر روی حرف های ادا شده تمرکز دارد و کمتر به جنبه های آهنگین توجه می کند. کلمات بیان شده می تواند یک گفتگوی ساده، یک شعر یا حتی بخشی از سخنرانی و پادکست باشد.

مقایسه ژانرهای موسیقی در یک کلیپ:

<https://www.aparat.com/v/iB5pQ>

برای بک گراند بازی می توانید از موسیقی با سبک Acoustic/Folk استفاده کنید:

<https://www.bensound.com/royalty-free-music/acoustic-folk>

مروری بر پیش نیازهای زبان جاوا اسکریپت:

<https://www.w3schools.com/html/exercise.asp>

https://www.w3schools.com/html/html_quiz.asp

https://www.w3schools.com/css/css_exercises.asp

https://www.w3schools.com/css/css_quiz.asp

PWA: Progressive Web Application

برنامه وب پیشرو چیست؟

Platform = بستر/سکو/پلتفرم

- بستر سخت افزاری

کامپیوتر دسکتاپ / لپ تاپ / تبلت / موبایل / کنسول های بازی / تلویزیون های هوشمند

- بستر نرم افزاری

ویندوز / لینوکس / مک / آندروید / iOS / سیستم های عامل های WearOS

Cross-Platform: برنامه یا بازی ای که بر روی همه بسترها اجرا شود، غیر وابسته به بستر گفته می شود.

Web: برای اجرا فقط به یک مرورگر نیاز دارد و مرورگر روی همه بسترها موجود است.

زبان‌های برنامه‌نویسی از لحاظ محل اجرای کد:

1 - زبان‌های سمت کلاینت (Client-side | Static | Front-end)

در این زبان‌ها، کدها ابتدا به سیستم کاربر دانلود و منتقل می‌شود و سپس آنجا اجرا می‌شود.

HTML, CSS, JavaScript

2 - زبان‌های سمت سرور (Server-side | Dynamic | Back-end)

در این زبان‌ها، اصل کدها بر روی سرور قرار دارد و آنجا اجرا می‌شود و فقط خروجی کدها به سیستم کاربر (کلاینت) منتقل می‌شود.

PHP, ASP, Python, NodeJS

برنامه‌نویس Fullstack:

یعنی هم به برنامه‌نویسی سمت کلاینت مسلط است و هم به برنامه‌نویسی سمت سرور. یعنی حداقل به HTML, CSS, JS, PHP|ASP|NodeJS مسلط است.

زبان جاوا اسکریپت

- زبان جاوا اسکریپت پرمخاطب‌ترین زبان برنامه‌نویسی دنیا است.
- یادگیری این زبان، آسان است.
- HTML برای نمایش محتوای (Display Content) صفحات وب استفاده می‌شود.
- CSS برای ساخت نمای سایت (Layout) استفاده می‌شود.
- JS برای برنامه‌نویسی رفتار سایت (Behavior) استفاده می‌شود.
- مرورگرها فقط و فقط سه زبان HTML و CSS و JS را می‌شناسند و اجرا می‌کنند.
- نیازی به دانلود چیزی برای اجرای جاوا اسکریپت نیست.
- همه مرورگرها به صورت built-in در داخل خود جاوا اسکریپت را پشتیبانی می‌کنند.
- جاوا اسکریپت رایگان است.
- زبان جاوا اسکریپت کاملاً با زبان جاوا متفاوت است.
- جاوا اسکریپت در سال ۱۹۹۵ توسط Brendan Eich ابداع و معرفی و در سال ۱۹۹۷ به یک استاندارد در W3C تبدیل شد.

با جاوا اسکریپت چه کارهایی می‌توان انجام داد؟

- JS می‌تواند محتوای یک تگ HTML را تغییر دهد. `<p>Content</p>`
- JS می‌توان مقدار صفات HTML را تغییر دهد. `<p align="center">Content</p>`
- JS می‌تواند CSS را تغییر دهد. `<p style="color:red;">Content</p>`

روش‌های درج کدهای JS در صفحه:

1 - روش inline (در خط):

در این روش، کدهای JS در یکی از رخدادهای تگ مورد نظر نوشته می‌شود.

```
<input type="button" onclick="alert('Hello World!')">
```

2 - روش Embedded (توکار):

در این روش، کدهای JS در بین تگ script معمولاً در بالای صفحه بین تگ head نوشته می‌شود.

```
...  
<head>
```

```

<meta charset="utf-8">
<title>JS</title>
<script>
    alert('Welcome...');
</script>

```

3- روش External (خارجی):

در این روش، کدهای JS در یک فایل خارجی با پسوند JS نوشته می شود و سپس توسط کد زیر به صفحاتی که لازم است، وارد می شود:

```

<script src="scripts.js"></script>

```

توجه: استفاده از روش سوم بر روش های دیگر اولویت دارد و استفاده از روش دوم، بر روش اول اولویت دارد. مزایای استفاده از روش سوم:

- 1- ایزوله شدن کدها: یعنی کدهای HTML در یک صفحه مجزا نوشته می شود، کدهای CSS نیز در یک صفحه مجزا و کدهای JS هم در یک صفحه مجزا.
- 2- خوانایی کدها بهتر می شود.
- 3- از تکرار کد در صفحات مختلف جلوگیری می شود.
- 4- فایل حاوی کدهای JS وقتی یک بار دانلود شد، روی مرورگر کاربر کش (Cache) می شود و در صفحات دیگر، آن فایل دانلود نمی شود و این، سرعت لود صفحات را بالا می برد.

نمایش در خروجی:

1- استفاده از :innerHTML

```

<input type="button" onclick="document.getElementById('test').innerHTML='Goodbye!'"
value="Click Here! ">
<div id="test">Hello!</div>

```

2- استفاده از :document.write()

```

<script>
    document.write('Have a good day!');
</script>

```

3- استفاده از :alert()

```

<input type="button" onclick="alert('Hello World!')" value="Click Here! ">

```

4- استفاده از :console.log()

```
<input type="button" onclick="console.log('Testing 1 2 3')" value="Click Here! ">
```

این روش مختص برنامه‌نویسان و برای انجام تست‌هایی در پشت صحنه سایت است؛ برای مثال

برای Debugging (خطایابی)

گرامر زبان جاوا اسکریپت (JS Syntax):

- این زبان به حروف کوچک و بزرگ حساس است (Case-sensitive است)؛ پس alert با Alert متفاوت است.
- ساختار کلی دستورات در این زبان، C-based (مبتنی بر C) است؛ یعنی به طور مثال عبارات شرطی در مقابل if و حلقه‌ها باید داخل پرانتز نوشته شوند، بلاک‌ها با {} مشخص می‌شوند...
- قرار دادن ; در انتهای دستورات اختیاری است اما بهتر است عادت کنید که قرار دهید.
- در این زبان، هر کجا یک فاصله لازم بود، درج کنید، بیش از یک فاصله مشکلی ایجاد نمی‌کند.
- بهتر است یک خط کد بیش از ۸۰ کاراکتر نباشد. اگر خواستید یک خط را بشکنید (Line break) بهترین جا برای شکستن، بعد از یک عملگر است؛ مثلاً بعد از =

```
document.getElementById("demo").innerHTML =  
"Hello Dolly!";
```

تعریف Statement

هر برنامه کامپیوتری (Program) از یک سری دستورالعمل (Instruction) تشکیل شده که باید اجرا شوند. در زبان‌های برنامه‌نویسی به این دستورالعمل‌ها در اصطلاح «عبارت» یا statement گفته می‌شود.

توضیحات در زبان جاوا اسکریپت

توضیحات در برنامه‌نویسی دو هدف دارد:

- 1 - نوشتن راهنما و توضیحاتی برای برنامه‌نویسان بعدی یا مراجعات بعدی خودمان
- 2 - از کار انداختن موقت یک قطعه کد

توضیحات در این زبان مانند زبان C است؛ یعنی دو نوع توضیحات داریم:

1 - Inline comment

```
// test
```

2 - Block comment

```
/*  
Test 1
```

Test 2

*/

نکته: عادت به نوشتن توضیحات در کدنویسی، عادت بسیار خوبی است.

متغیرها¹ در زبان جاوا اسکریپت

متغیرها در واقع مخزنی هستند برای نگهداری موقت داده‌ها. این مخزن‌ها بر روی RAM قرار می‌گیرند و پس از اتمام اجرای برنامه، حذف می‌شوند. محتوای متغیرها ممکن است در طول برنامه چندین بار تغییر کند.

در زبان JS هنگام تعریف اولیه² متغیرها، نوع داده بیان نمی‌شود. برای همه نوع داده، از کلمه کلیدی var استفاده می‌کنیم؛ مثال:

```
var x = 5;  
var y = 3.5;  
var z = "Test";
```

جاوا اسکریپت از روی محتوای متغیر به نوع آن پی می‌برد. یعنی اگر در یک متغیر برای مثال عدد ۵ بریزید، جاوا اسکریپت آن را int در نظر می‌گیرد و اگر ۳.۵ بریزید، float در نظر می‌گیرد و...

قواعد نامگذاری متغیرها

به نام متغیرها در اصطلاح identifier یا شناسه آن گفته می‌شود.

1 - فقط حروف کوچک و بزرگ انگلیسی، اعداد و کاراکتر _ در نام متغیر مجاز است.

مثال غلط:

```
var x.2 = 5;
```

x

2 - نام متغیر نباید با عدد شروع شود.

3 - نام متغیر می‌تواند با کاراکتر \$ شروع شود.

4 - زبان جاوا اسکریپت به حروف کوچک و بزرگ حساس است.

5 - کلمات رزرو شده در زبان جاوا اسکریپت نمی‌توانند به عنوان نام متغیر استفاده شوند.

مثال غلط:

```
var for = 3;
```

x

¹ Variables

² Declaration

ثوابت^۳ در زبان جاوا اسکریپت

ثوابت نیز مانند متغیرها مخزنی برای نگهداری داده هستند؛ با این تفاوت که مقدار آنها در طول اجرای برنامه قابل تغییر نیست.

نحوه تعریف اولیه ثوابت:

```
const PI = 3.141592653589793;
```

عملگرها^۴ در زبان جاوا اسکریپت

عملگرها در زبان JS بسیار شبیه به زبان C هستند؛ با چند تفاوت کوچک: در این زبان، عملگر + برای ترکیب رشته‌ای نیز استفاده می‌شود. عملگر ** برای محاسبه توان در نظر گرفته شده است.

عملگرهای ریاضی (Arithmetic Operators):

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

عملگرهای انتساب (Assignment Operators):

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
**=	x **= y	x = x ** y

عملگر رشته‌ای (String Operators):

برای اتصال دو رشته به یکدیگر از عملگر + استفاده می‌شود؛ مثال:

```
var txt1 = "Hamid Reza";  
var txt2 = "Niroomand";  
var txt3 = txt1 + " " + txt2;
```

عملگرهای مقایسه‌ای (Comparison Operators):

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type

³ Constants

⁴ Operators

>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

تفاوت == با === در این است که اولی فقط بررسی می کند که آیا محتوای دو طرف تساوی با هم برابر است یا خیر؟ اما دومی علاوه بر محتوا، «نوع داده» دو طرف تساوی را نیز بررسی می کند.

```
var x = 5;
var y = "5";
alert(x==y);
alert(x===y);
```

خروجی اول برابر با true خواهد بود اما خروجی دوم برابر با false

عملگرهای منطقی (Logical Operators):

Operator	Description
&&	logical and
	logical or
!	logical not

اولویت عملگرها

برای درک اولویت عملگرها، بررسی کنید در کد زیر چه مقداری در X قرار خواهد گرفت؟

```
var x = 100 + 50 * 3;
```

آیا ۴۵۰ در X قرار خواهد گرفت یا ۲۵۰؟

اولویت اجرای عملگرها در زبان JS به ترتیب زیر است:

Value	Operator	Description	Example
20	()	Expression grouping	(3 + 4)
19	.	Member	person.name
19	[]	Member	person["name"]
19	()	Function call	myFunction()
19	new	Create	new Date()
17	++	Postfix Increment	i++
17	--	Postfix Decrement	i--
16	++	Prefix Increment	++i
16	--	Prefix Decrement	--i
16	!	Logical not	!(x==y)
16	typeof	Type	typeof x
15	**	Exponentiation (ES2016)	10 ** 2

14	*	Multiplication	10 * 5
14	/	Division	10 / 5
14	%	Division Remainder	10 % 5
13	+	Addition	10 + 5
13	-	Subtraction	10 - 5
12	<<	Shift left	x << 2
12	>>	Shift right	x >> 2
12	>>>	Shift right (unsigned)	x >>> 2
11	<	Less than	x < y
11	<=	Less than or equal	x <= y
11	>	Greater than	x > y
11	>=	Greater than or equal	x >= y
11	in	Property in Object	"PI" in Math
11	instanceof	Instance of Object	instanceof Array
10	==	Equal	x == y
10	===	Strict equal	x === y
10	!=	Unequal	x != y
10	!==	Strict unequal	x !== y
9	&	Bitwise AND	x & y
8	^	Bitwise XOR	x ^ y
7		Bitwise OR	x y
6	&&	Logical AND	x && y
5		Logical OR	x y
4	? :	Condition	? "Yes" : "No"
3	+=	Assignment	x += y
3	/=	Assignment	x /= y
3	-=	Assignment	x -= y
3	*=	Assignment	x *= y
3	%=	Assignment	x %= y
3	<<=	Assignment	x <<= y
3	>>=	Assignment	x >>= y
3	>>>=	Assignment	x >>>= y
3	&=	Assignment	x &= y
3	^=	Assignment	x ^= y
3	=	Assignment	x = y
2	yield	Pause Function	yield x
1	,	Comma	5 , 6

تمرین:

بررسی کنید خروجی کدهای زیر چه خواهد بود؟

```
var x = 16 + 4 + "Volvo";
alert(x);
```

```
var x = "Volvo" + 16 + 4;
alert(x);
```

تابع در زبان جاوا اسکریپت

توابع در زبان‌های برنامه‌نویسی یک نوع **پیمانه** به حساب می‌آیند که برای **جلوگیری از تکرار**، به کار گرفته می‌شوند.

ساختار تعریف تابع در زبان جاوا اسکریپت:

```
function name(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

مثال: تابعی که نام و نام خانوادگی را دریافت می‌کند و با تزیینات خاصی در یک **div** نمایش می‌دهد:

```
<p id="demo"></p>
```

```
<script>
function welcome(fname, lname) {
  return '<div style="color: green;">Welcome, '+fname+' '+lname+'</div>';
}
document.getElementById("demo").innerHTML = welcome("Ali", "Hasani");
</script>
```

نکته: در لیست پارامترهای ورودی نیازی به آوردن کلمه **var** نیست.

تابع setInterval

این تابع، یک تابع را به عنوان پارامتر اول دریافت می‌کند و یک فاصله زمانی^۵ (به میلی‌ثانیه) را به عنوان پارامتر دوم؛ سپس آن تابع را مکرراً در فواصل زمانی مشخص شده تکرار می‌کند.

```
setInterval(function() {
  //code to repeat
}, interval);
```

مثال: کدی بنویسید که هر ۱ ثانیه یک بار، عددی را که از ۱ شروع می‌شود، یک واحد افزایش دهد و در یک **div** چاپ کند.

```
<script>
var x = 1;
setInterval(function() {
  document.getElementById('demo').innerHTML = x;
  x++;
}, 1000);
```

⁵ Interval


```
</script>  
<div id="demo"></div>
```

تابع setTimeout

این تابع، یک تابع را به عنوان پارامتر اول دریافت می‌کند و یک زمان انتظار^۶ (به میلی‌ثانیه) را به عنوان پارامتر دوم؛ سپس آن تابع را پس از منتظر ماندن به میزان بیان شده، اجرا می‌کند.

```
setTimeout(function() {  
    //code to run  
}, timeout);
```

مثال: کدی بنویسید که پس از گذشت ۳ ثانیه از ورود کاربر، پیغام «خوش آمدید» را نمایش دهد.

```
<script>  
setTimeout(function() {  
    alert('Welcome..');  
}, 3000);  
</script>
```

اولین بازی:

بازی پرش از روی مانع

⁶ Time Out

شیء یا Object در زبان جاوا اسکریپت

در دنیای واقعی آنچه قابل لمس توسط یکی از حس‌های شما باشد، یک شیء نامیده می‌شود؛ مانند: ماشین، ساختمان، میوه، کتاب و...

مفهوم شیء به برنامه‌نویسی وارد شد تا درک مفاهیم برنامه‌نویسی را آسان‌تر کند.

هر شیء یک سری صفات یا ویژگی‌ها دارد که در برنامه‌نویسی به آن‌ها Properties گفته می‌شود.

و هر شیء یک سری عملکرد دارد که در برنامه‌نویسی به آن‌ها Methods گفته می‌شود.

مثلاً یک ماشین، صفاتی مانند رنگ، نوع، سال تولید، طول، عرض، ارتفاع و... دارد و عملکردهایی مانند «استارت»، «رانده شدن»، توقف و...

مفهوم Encapsulation (کپسوله‌سازی)

یک شیء در حقیقت یک کپسول است که در داخل خود، صفات و عملکردهای مربوط به خود را حمل می‌کند.

نحوه تعریف یک شیء با صفات و عملکردهایش در جاوا اسکریپت

```
var person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
```

مثال: برنامه‌ای بنویسید که نام، نام خانوادگی، نام پدر، سال تولد، قد، معدل کارشناسی را از ورودی دریافت کند و در یک شیء به نام student ذخیره کند و آن شیء متودی به نام Print داشته باشد که هنگام فراخوانی، این صفات را در یک div با آی‌دی st_prop نمایش دهد.

```
var st;
function AddStudent() {
  st={
    firstName:prompt("نام خود را وارد نمایید"),
    lastName:prompt("نام خانوادگی خود را وارد نمایید"),
    fatherName:prompt("نام پدر خود را وارد نمایید"),
    birthdate:prompt("سال تولد خود را وارد نمایید"),
    height:prompt("قد خود را وارد نمایید"),
    ave:prompt("معدل دوره کارشناسی خود را وارد نمایید"),
    Print:function() {
      document.getElementById('st_prop').innerHTML=
        "نام: "+this.firstName+
        "<br>نام خانوادگی: "+this.lastName+
        "<br>نام پدر: "+this.fatherName+
```

```
    "<br>سال تولد: "+this.birthdate+
    "<br>قد: "+this.height+
    "<br>معدل دوره کارشناسی: "+this.ave
  }
}
```

کلمه کلیدی this

برای دسترسی به مقدار یک صفت از یک شیء در داخل متودهای آن شیء از کلمه کلیدی this استفاده می شود:

this.firstName

آرایه‌ها در زبان جاوا اسکریپت

آرایه متغیری است که به جای یک مقدار، می‌تواند چندین مقدار را در خود ذخیره و حمل کند.

آرایه‌ها از پرکاربردترین **Data Type**ها در همه زبان‌های برنامه‌نویسی هستند.

آرایه را می‌تواند معادل یک ساختمان آپارتمانی در نظر گرفت. هر واحد آپارتمان در بحث آرایه در اصطلاح **Element** نامیده می‌شود. شماره خانه‌های آرایه در اصطلاح اندیس یا **index** نامیده می‌شوند. اندیس‌ها از 0 شروع می‌شوند.

مهم‌ترین دلیل استفاده از آرایه این است که خانه‌های آرایه شماره‌ترتیبی دارند و می‌توان آن‌ها را با یک حلقه پیمایش کرد.

مثالی از تعریف یک آرایه:

```
var cars = ["Saab", "Volvo", "BMW"];
```

دسترسی به مقدار یک خانه یا المنت:

```
var name = cars[0];
```

در زبان جاوا اسکریپت اندیس‌های آرایه‌ها نمی‌توانند نام داشته باشند. اگر قرار است به اندیس‌ها نام بدهید، باید از شیء استفاده کنید.

صفت length

با فراخوانی صفت **length** روی نام آرایه، تعداد عناصر آن به دست می‌آید؛ مثال: خروجی کد زیر چند خواهد بود؟

```
alert(cars.length);
```

افزودن عنصر جدید به آرایه

از تابع **push** برای این کار استفاده کنید:

```
cars.push("Pride");
```

متود pop

این متود، عنصر آخر آرایه را حذف می‌کند. اگر در مقابل نام یک متغیر فراخوانی کنید، علاوه بر حذف، آن عنصر را به متغیر نامبرده شده نیز نسبت می‌دهد:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
var x = fruits.pop();
```

متود shift

این متود مانند pop عمل می کند با این تفاوت که از ابتدای آرایه عنصر را حذف می کند.

متود unshift

این متود یک عنصر را به ابتدای آرایه اضافه می کند؛ مثال:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.unshift("Lemon");
```

متود concat

از این متود برای ادغام دو آرایه استفاده می شود؛ مثال: در آرایه myChildren چه مقادیری قرار خواهد گرفت؟

```
var myGirls = ["Cecilie", "Lone"];
var myBoys = ["Emil", "Tobias", "Linus"];
var myChildren = myGirls.concat(myBoys);
```

متود slice

از این متود برای جدا کردن بخشی از آرایه از آن استفاده می شود.

```
var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
var citrus = fruits.slice(1, 3);
```

خط دوم کد بالا، عناصر از خانه شماره ۱ تا خانه شماره ۳ (و البته منهای خود آن خانه) را در آرایه citrus

می ریزد. یعنی خروجی کد بالا چه خواهد بود؟

متود sort

رخدادها در زبان جاوا اسکریپت

رخداد، اتفاقی است که بر روی یک عنصر در صفحه وب انجام می‌گیرد. جاوا اسکریپت می‌تواند به این اتفاقات واکنش نشان دهد.

چند نمونه رخداد:

- کاربر روی یک عنصر کلیک کند.
- کاربر یک عنصر را drag کند.
- بارگذاری صفحه کامل شود.
- ...

لیستی از رخدادهایی که جاوا اسکریپت پشتیبانی می‌کند:

Event	Description
abort	The event occurs when the loading of a media is aborted
afterprint	The event occurs when a page has started printing, or if the print dialogue box has been closed
animationend	The event occurs when a CSS animation has completed
animationiteration	The event occurs when a CSS animation is repeated
animationstart	The event occurs when a CSS animation has started
beforeprint	The event occurs when a page is about to be printed
beforeunload	The event occurs before the document is about to be unloaded
blur	The event occurs when an element loses focus
canplay	The event occurs when the browser can start playing the media (when it has buffered enough to begin)
canplaythrough	The event occurs when the browser can play through the media without stopping for buffering
change	The event occurs when the content of a form element, the selection, or the checked state have changed (for <input>, <select>, and <textarea>)
click	The event occurs when the user clicks on an element
contextmenu	The event occurs when the user right-clicks on an element to open a context menu
copy	The event occurs when the user copies the content of an element

cut	The event occurs when the user cuts the content of an element
dblclick	The event occurs when the user double-clicks on an element
drag	The event occurs when an element is being dragged
dragend	The event occurs when the user has finished dragging an element
dragenter	The event occurs when the dragged element enters the drop target
dragleave	The event occurs when the dragged element leaves the drop target
dragover	The event occurs when the dragged element is over the drop target
dragstart	The event occurs when the user starts to drag an element
drop	The event occurs when the dragged element is dropped on the drop target
durationchange	The event occurs when the duration of the media is changed
ended	The event occurs when the media has reach the end (useful for messages like "thanks for listening")
error	The event occurs when an error occurs while loading an external file
focus	The event occurs when an element gets focus
focusin	The event occurs when an element is about to get focus
focusout	The event occurs when an element is about to lose focus
fullscreenchange	The event occurs when an element is displayed in fullscreen mode
fullscreenerror	The event occurs when an element can not be displayed in fullscreen mode
hashchange	The event occurs when there has been changes to the anchor part of a URL
input	The event occurs when an element gets user input
invalid	The event occurs when an element is invalid
keydown	The event occurs when the user is pressing a key
keypress	The event occurs when the user presses a key
keyup	The event occurs when the user releases a key
load	The event occurs when an object has loaded
loadeddata	The event occurs when media data is loaded
loadedmetadata	The event occurs when meta data (like dimensions and duration) are loaded

loadstart	The event occurs when the browser starts looking for the specified media
message	The event occurs when a message is received through the event source
mousedown	The event occurs when the user presses a mouse button over an element
mouseenter	The event occurs when the pointer is moved onto an element
mouseleave	The event occurs when the pointer is moved out of an element
mousemove	The event occurs when the pointer is moving while it is over an element
mouseover	The event occurs when the pointer is moved onto an element, or onto one of its children
mouseout	The event occurs when a user moves the mouse pointer out of an element, or out of one of its children
mouseup	The event occurs when a user releases a mouse button over an element
mousewheel	Deprecated. Use the wheel event instead
offline	The event occurs when the browser starts to work offline
online	The event occurs when the browser starts to work online
open	The event occurs when a connection with the event source is opened
pagehide	The event occurs when the user navigates away from a webpage
pageshow	The event occurs when the user navigates to a webpage
paste	The event occurs when the user pastes some content in an element
pause	The event occurs when the media is paused either by the user or programmatically
play	The event occurs when the media has been started or is no longer paused
playing	The event occurs when the media is playing after having been paused or stopped for buffering
popstate	The event occurs when the window's history changes
progress	The event occurs when the browser is in the process of getting the media data (downloading the media)
ratechange	The event occurs when the playing speed of the media is changed
resize	The event occurs when the document view is resized
reset	The event occurs when a form is reset

scroll	The event occurs when an element's scrollbar is being scrolled
search	The event occurs when the user writes something in a search field (for <code><input="search"></code>)
seeked	The event occurs when the user is finished moving/skipping to a new position in the media
seeking	The event occurs when the user starts moving/skipping to a new position in the media
select	The event occurs after the user selects some text (for <code><input></code> and <code><textarea></code>)
show	The event occurs when a <code><menu></code> element is shown as a context menu
stalled	The event occurs when the browser is trying to get media data, but data is not available
<code>storage</code>	The event occurs when a Web Storage area is updated
submit	The event occurs when a form is submitted
suspend	The event occurs when the browser is intentionally not getting media data
timeupdate	The event occurs when the playing position has changed (like when the user fast forwards to a different point in the media)
toggle	The event occurs when the user opens or closes the <code><details></code> element
touchcancel	The event occurs when the touch is interrupted
touchend	The event occurs when a finger is removed from a touch screen
touchmove	The event occurs when a finger is dragged across the screen
touchstart	The event occurs when a finger is placed on a touch screen
transitionend	The event occurs when a CSS transition has completed
unload	The event occurs once a page has unloaded (for <code><body></code>)
volumechange	The event occurs when the volume of the media has changed (includes setting the volume to "mute")
waiting	The event occurs when the media has paused but is expected to resume (like when the media pauses to buffer more data)
wheel	The event occurs when the mouse wheel rolls up or down over an element

رشته‌ها در جاوا اسکریپت (String)

- تفاوتی ندارد که دور مقدار متغیر رشته‌ای را با دابل کوتیشن یا سنگل کوتیشن احاطه کنید. پس این دو خط کد یکسان هستند:

```
var carName1 = "Volvo XC60"; // Double quotes
var carName2 = 'Volvo XC60'; // Single quotes
```

- صفت length اگر بر روی متغیر رشته‌ای اعمال شود، طول رشته را برمی‌گرداند:

```
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
var sln = txt.length; //26
```

متودهای مربوط به رشته‌ها

- متود indexOf

- از این متود برای یافتن یک رشته در یک رشته دیگر استفاده می‌شود؛ مثال: در کد زیر چه عددی در pos قرار خواهد گرفت؟ (پاسخ: ۷)

```
var str = "Please locate where 'locate' occurs!";
var pos = str.indexOf("locate");
```

توجه: این تابع، موقعیت اولین مشاهده‌ی رشته مورد جستجو را برمی‌گرداند.

- متود lastIndexOf

- این متود، موقعیت آخرین مشاهده‌ی رشته مورد جستجو در رشته اولیه را برمی‌گرداند؛ مثال: در کد زیر چه عددی در pos قرار خواهد گرفت؟ (پاسخ: ۲۱)

```
var str = "Please locate where 'locate' occurs!";
var pos = str.lastIndexOf("locate");
```

- توجه: هر دو تابع indexOf و lastIndexOf در صورتی که رشته مورد نظر را پیدا نکنند، عدد -1 برمی‌گردانند.

- توجه: اگر در متود indexOf به عنوان پارامتر دوم، یک عدد وارد کنید، این متود از آن نقطه به بعد دنبال رشته مورد نظر می‌گردد؛ مثال:

```
var str = "Please locate where 'locate' occurs!";
var pos = str.indexOf("locate", 15);
```

- متود search

- این متود نیز مانند `indexOf` عمل می کند. با این تفاوت که پارامتر دوم به خود نمی گیرد. اما در عوض این متود یک برتری نسبت به `indexOf` دارد و آن اینکه عبارات منظم (Regular Expressions)^۷ را پشتیبانی می کند.

- متود `slice`

- از این متود برای جدا کردن یک زیررشته از یک رشته بزرگ تر استفاده می شود؛ مثال: خروجی قطعه کد زیر چه خواهد بود؟ (پاسخ: Banana)

```
var str = "Apple, Banana, Kiwi";
var res = str.slice(7, 13);
alert(res);
```

- توجه: پارامتر اول در این تابع، نقطه شروع جداسازی و پارامتر دوم نقطه پایان است.
- توجه: اگر پارامتر دوم بیان نشود، متود، تا انتهای رشته پیش می رود.
- توجه: اگر عدد اول به صورت منفی وارد شود، شمارش کاراکترها از **انتهای رشته** صورت می گیرد.

- متود `substring`

- این متود نیز دقیقاً مانند `slice` عمل می کند با این تفاوت که متود `substring` اعداد منفی قبول نمی کند.

- متود `substr`

- این متود نیز مانند `slice` عمل می کند با این تفاوت که در پارامتر دوم به جای نقطه پایان جداسازی، طول رشته ای که باید جدا شود را بیان می کنیم؛ مثال: خروجی قطعه کد زیر چه خواهد بود؟ (پاسخ: Ban)

```
var str = "Apple, Banana, Kiwi";
var res = str.substr(7, 3);
alert(res);
```

- توجه: ترجیحاً از متود `substr` برای جداسازی استفاده کنید.

- متود `replace`

- این متود دو پارامتر ورودی دارد؛ پارامتر اول رشته ای است که باید پیدا شود و پارامتر دوم رشته ای است که باید جایگزین رشته پیدا شده شود؛ مثال: خروجی قطعه کد زیر چه خواهد بود؟ (پاسخ: Please visit W3Cshools!)

```
str = "Please visit Microsoft!";
```

^۷ عبارات منظم، یک الگو برای جستجو مشخص می کنند؛ مثلاً فرض کنید دنبال رشته هایی با الگوی ایمیل (`x@y.z`) می گردید. در این حالت باید از عبارات منظم استفاده کنید.

```
var n = str.replace("Microsoft", "W3Schools");  
alert(n);
```

- متود toUpperCase

- این متود همه حروف رشته را به حروف بزرگ انگلیسی تبدیل می کند؛ مثال: خروجی قطعه کد زیر چه خواهد بود؟ (پاسخ: HAMID REZA)

```
var x = "Hamid Reza";  
var x = x.toUpperCase();  
alert(x);
```

- متود toLowerCase

- این متود همه حروف را به حروف کوچک انگلیسی تبدیل می کند.

- متود trim

- این متود فضای سفید ابتدا و انتهای رشته را حذف می کند.

- متود split

- این متود برای تبدیل رشته به آرایه استفاده می شود؛ مثال: در قطعه کد زیر، تابع split، رشته x را با جداکننده کاما (,) از هم جدا کرده و هر بخش را در یک خانه از یک آرایه ذخیره می کند.

```
var txt = "a,b,c,d,e";  
var ar = txt.split(",");  
alert(ar[2]);
```

- خروجی قطعه کد بالا چیست؟ (پاسخ: C)

- برخورد با رشته به صورت آرایه

- می توانید یک رشته را یک آرایه تصور کنید؛ بنابراین برای دسترسی به یک کاراکتر در موقعیت خاص، می توانید شماره آن موقعیت را داخل براکت بیان کنید؛ مثال: خروجی قطعه کد زیر چه خواهد بود؟ (پاسخ: R)

```
var x = "Hamid Reza";  
alert(x[6]);
```

تمرین: بازی ZType

متودهای مربوط به اعداد در جاوا اسکریپت

- متود toString

- این متود، متغیر عددی را تبدیل به متغیر رشته‌ای می‌کند؛ مثال: کد زیر متغیر X را از حالت عددی به حالت رشته‌ای تغییر می‌دهد.

```
var x = 123;  
x.toString();
```

- متود toFixed

- این متود تعداد ارقام بعد از اعشار را مشخص می‌کند. اگر تعداد ارقام بیان شده کمتر از ارقام عدد باشد، عدد گرد می‌شود.

```
var x = 9.656;  
x.toFixed(0); // returns 10  
x.toFixed(2); // returns 9.66  
x.toFixed(4); // returns 9.6560  
x.toFixed(6); // returns 9.656000
```

- متود toPrecision

- این متود تعداد ارقام کل عدد (صحیح + اعشاری) را مشخص می‌کند.

```
var x = 9.656;  
x.toPrecision(); // returns 9.656  
x.toPrecision(2); // returns 9.7  
x.toPrecision(4); // returns 9.656  
x.toPrecision(6); // returns 9.65600
```

- متودهای تبدیل متغیر رشته‌ای به عددی:

- متود parseInt

- این متود، محتوای رشته‌ای یک متغیر را (تا جایی که ممکن باشد) به عدد صحیح تبدیل می‌کند.

```
parseInt("10"); // returns 10  
parseInt("10.33"); // returns 10  
parseInt("10 20 30"); // returns 10  
parseInt("10 years"); // returns 10  
parseInt("years 10"); // returns NaN
```

- متود parseFloat

- این متود، محتوای رشته‌ای یک متغیر را (تا جایی که ممکن باشد) به عدد اعشاری تبدیل می‌کند.

```
parseFloat("10"); // returns 10  
parseFloat("10.33"); // returns 10.33  
parseFloat("10 20 30"); // returns 10  
parseFloat("10 years"); // returns 10  
parseFloat("years 10"); // returns NaN
```

- تمرین: ماشین حساب

مبحث تاریخ و زمان در جاوا اسکریپت

برای کار با تاریخ در جاوا اسکریپت باید ابتدا یک شیء از کلاس Date() ایجاد شود:

```
var d = new Date();
```

از آنجا که جاوا اسکریپت یک زبان سمت کلاینت است، به طور پیش فرض تاریخ سیستم کاربر را ملاک عمل قرار می دهد.

سازندهی Date به چهار صورت قابل فراخوانی است:

```
new Date()  
new Date(year, month, day, hours, minutes, seconds, milliseconds)  
new Date(milliseconds)  
new Date(date string)
```

در حالت اول: لحظه ی فعلی، ملاک عمل قرار خواهد گرفت؛ مثال: قطعه کد زیر، شماره ی روز فعلی (امروز) را برمی گرداند:

```
var d = new Date();  
alert(d.getDay());
```

در حالت دوم: تاریخ بیان شده، ملاک عمل قرار خواهد گرفت؛ مثال: قطعه کد زیر، شماره روز 1 آپریل 2009 را برمی گرداند:

```
var d = new Date(2009,4,1,12,0,0,0);  
alert(d.getDay());
```

در حالت سوم: میزان زمان (میلی ثانیه) گذشته از تاریخ 1970/01/01 00:00:00، ملاک عمل قرار خواهد گرفت؛ مثال: قطعه کد زیر، شماره روز مربوط به یک میلیارد میلی ثانیه که از تاریخ مذکور گذشته را بیان می کند:

```
var d = new Date(1000000000000);  
alert(d.getDay());
```

در حالت سوم: تاریخ بیان شده به صورت رشته ای، ملاک عمل قرار می گیرد؛ مثال: قطعه کد زیر، شماره روز مربوط به تاریخ عملیات ۱۱ سپتامبر را مشخص می کند:

```
var d = new Date("Sep 11, 2001 08:46:00");  
alert(d.getDay());
```

متودهای کار با تاریخ

بر روی شیء d در مثال های بالا می توان متودهای زیر را اجرا کرد:

Method	Description
getFullYear()	Get the year as a four digit number (yyyy)
getMonth()	Get the month as a number (0-11)
getDate()	Get the day as a number (1-31)
getHours()	Get the hour (0-23)
getMinutes()	Get the minute (0-59)
getSeconds()	Get the second (0-59)
getMilliseconds()	Get the millisecond (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

تمرین: یک مبدل تاریخ میلادی به شمسی و برعکس بسازید.

مبحث Math در جاوا اسکریپت

- Math.PI عدد پی را در خود دارد؛ مثال: قطر یک دایره را از ورودی دریافت کنید و مساحت و محیط آن را محاسبه و چاپ کنید.

```
var d = prompt("قطر دایره را وارد نمایید");  
var a = d/2*d/2*Math.PI;  
var p = d*Math.PI;  
alert("مساحت: "+a);  
alert("محیط: "+p);
```

- Math.round(x) عدد را گرد می کند.
- Math.ceil(x) سقف عدد را به دست می آورد؛ مثال: Math.round(۳.۱) عدد ۴ خواهد شد.
- Math.floor(x) کف عدد را به دست می آورد؛ مثال: Math.round(۳.۹) عدد ۳ خواهد شد.
- Math.pow(x,y) به توان x ی را محاسبه خواهد کرد.
- Math.sqrt(x) جذر x را محاسبه خواهد کرد.
- Math.abs(x) قدر مطلق x را محاسبه خواهد کرد.
- Math.log2(x) لوگاریتم x را در مبنای ۲ محاسبه خواهد کرد.
- Math.log10(x) لوگاریتم x در مبنای ۱۰ محاسبه خواهد کرد.
- برای به دست آوردن یک عدد رندوم مثلاً بین ۱ و ۱۰۰ از کد زیر استفاده نمایید:

```
Math.floor(Math.random() * 100) + 1;
```

تمرین: یک بازی برای تقویت ریاضی کاربر بسازید. به این صورت که یک عدد رندوم مثلاً بین ۱ تا ۱۰۰۰ به کاربر نشان دهید؛ سپس جذر، توان دوم، لوگاریتم در مبنای ۲ و ۱۰ و گرد-شده آن عدد و... را با حالتی بازی گونه از او بپرسید...

عبارات شرطی (Conditional Statements)

جاوا اسکریپت مانند اکثر زبان‌های برنامه‌نویسی از ساختار `if ... else` و `switch` پشتیبانی می‌کند.

ساختار `if ... else`

این ساختار به سه شکل دیده می‌شود:

<pre>if (condition) { //if body }</pre>	<pre>if (condition) { //if body } else { //else body }</pre>	<pre>if (condition1) { //if condition1 is true } else if (condition2) { //if condition2 is true } else if (condition3) { //if condition3 is true } . . . else { //if none of the conditions are true }</pre>
---	--	--

تمرین: برنامه Word Scramble

قِ وَالْقُرْآنِ الْمَجِيدِ

بِئَاتِي الْعَالَمِينَ وَالْقُرْآنِ الْمَجِيدِ عَجَبُوا نَسِيءَ جَاءَهُمْ قِ عَجِيبٌ فَقَالَ مِنْهُمْ مُنِيرٌ هَذَا (٢) أَنْ (١) بِلْ

معرفی سایت

سایت gamee.com مملو از بازی‌های آنلاین ساده است که می‌توانید از آن‌ها برای ساخت بازی الگو بگیرید.

Games for you

Every game with more missions



Karate Kido 2



Color Hit



Space Orbit



Getaway

<https://www.gamee.com> Basket Boy

Space Traveler



Stair Master 3D



Geometry Run 3D



Red and Blue



Sunshine Solitaire

دستور switch

در صورتی که یک قطعه کد به صورت `if ... else if ... else if ...` های متوالی پیاده سازی شده باشد و البته در صورتی که شرط بررسی شده در آنها به صورت تساوی (`==`) باشد، می توان از دستور `switch` استفاده کرد.

ساختار کلی این دستور به شکل زیر است:

```
switch(expression) {  
  case x:  
    // code block  
    break;  
  case y:  
    // code block  
    break;  
  default:  
    // code block  
}
```

مثال: بازی تاس چرخنده (Dive Roller)

یک تاس شش وجهی را بیندازی و به صورت شانسی (رندوم) یک وجه را نمایش دهید.

راهنما با عبارت `site:codepen.io dice css` با کمک گوگل در سایت `codepen.io` جستجو کنید و از

تاس هایی که دیگران طراحی کرده اند ایده بگیرید.

نظریه Levels of Processing

نظریه «سطوح پردازش» بیان می‌کند که هر چقدر روی یک چیز در هنگام به‌خاطر سپاری بیشتر پردازش انجام داده باشید، به‌یادآوری آن آسان‌تر خواهد بود.

حلقه‌ها در زبان جاوا اسکریپت

جاوا اسکریپت مانند زبان C سه حلقه مشهور `while` و `do ... while` و `for` را پشتیبانی می‌کند.

جدول زیر نشان می‌دهد که چگونه می‌توان جمع اعداد ۱ تا ۱۰۰ را با هر سه حلقه محاسبه کرد:

<pre>var i=1,sum=0; while(i<=100) { sum+=i; i++; } alert(sum);</pre>	<pre>var i=1,sum=0; do{ sum+=i; i++; }while(i<=100); alert(sum);</pre>	<pre>var sum=0; for(var i=1;i<=100;i++) { sum+=i; } alert(sum);</pre>
---	---	--

از حلقه `for in` برای پیمایش یک شیء (Object) یا گاهی برای پیمایش آرایه استفاده می‌شود. مثال زیر،

صفات شیء `person` را چاپ می‌کند:

```
var person = {fname:"John", lname:"Doe", age:25};
```

```
var text = "";
var x;
for (x in person) {
    text += person[x];
}
alert(text);
```

فریم‌ورک‌های جاوا اسکریپت

فریم‌ورک‌ها در حقیقت قطعه کدهای آماده هستند که بخشی از زحمت کدنویس را کم می‌کنند. کاری که در حالت عادی با چندین خط کد به زبان جاوا اسکریپت انجام می‌شود، با کمک فریم‌ورک‌ها با یک خط کد قابل انجام است.

یکی از مهم‌ترین دلایل استفاده از فریم‌ورک‌ها این است که این فریم‌ورک‌ها طوری نوشته شده‌اند که نتیجه دلخواه شما را بر روی تمامی مرورگرها به یک شکل نمایش دهند. (cross-browser یا browser compatibility)

برای جاوا اسکریپت فریم‌ورک‌های مختلفی وجود دارد که مشهورترین‌های آن‌ها عبارتند از:

- jQuery
- Angular.js
- React
- NodeJS
- Vue.js

جی کوئری (jQuery)

کارهایی که می‌توان با جی کوئری انجام داد:

- دستکاری تگ‌ها و صفات آن‌ها
- دستکاری CSS
- پشتیبانی از تمام رخدادهای HTML
- ایجاد افکت‌های زیبا و انیمیشن
- کار با AJAX
- ابزارهای کاربردی که بر اساس جی کوئری نوشته شده.

نحوه اضافه کردن jQuery به صفحه

1 - ابتدا این کتابخانه از سایت رسمی آن یعنی jquery.com دانلود کنید و در یکی از پوشه‌های سایت

خود قرار دهید.

2 - توسط قطعه کد زیر، jQuery را به صفحه دلخواه خود اضافه کنید:

```
<script src="js/jquery.min.js"></script>
```

آغاز کار با جی کوئری:

کدهای جی کوئری اکثراً بین قطعه کد زیر نوشته می شود:

```
$(function () {  
    //Code here  
});
```

این قطعه کد به مرورگر می فهماند که وقتی همه عناصر صفحه دانلود شدند و آماده اجرا شدی، کدهای داخل {} را اجرا کن.

Selectors در جی کوئری

خوشبختانه جی کوئری همان انتخابگرهای CSS را به عنوان انتخابگرهای خود در نظر گرفته است؛ بنابراین، اگر شما انتخابگرهای CSS را می شناسید، نیازی به یادگیری مبحث جدیدی ندارید.

مهم ترین انتخابگرهای CSS

- انتخاب همه عناصر:

```
*{  
}
```

- انتخاب تگ هایی با نام دلخواه:

```
a{  
}
```

- انتخاب چند عنصر با کلاس دلخواه:

```
.container{  
}
```

- انتخاب یک عنصر با آی دی خاص:

```
#logo{  
}
```

- انتخاب عناصر در آدرسی خاص:

```
#main_menu li a{  
}
```

در جی کوئری کافیست انتخابگرها را به این شکل انتخاب و متود دلخواه خود را بر روی آن اعمال کنید:

```
$("selector").x();
```

به جای X در قطعه کد بالا دو مورد قابل استفاده است:

1 - رخداد؛ مانند:

`click , dblclick , mouseenter , mouseleave , mousedown , mouseup ,
hover , focus , blur , ...`

2 - متوذهای جی کوئری؛ مانند:

`hide , show , toggle , animate , css , append , prepend , ..`

مثال: قطعه کد زیر مشخص می کند که اگر روی عنصری با آی دی `test` کلیک شد، عنصری با آی دی `demo` مخفی شود:

```
$(document).ready(function() {  
  $("#test").click(function() {  
    $("#demo").hide();  
  });  
});
```

رخدادها همان رخدادهایی که است که پیش از این در صفحات قبل آموزش داده شد؛ بنابراین، شما کافی ست توابع جی کوئری را یاد بگیرید.

متودهای مخفی/نمایان کردن عناصر

برای مخفی کردن عناصر از متود `hide` استفاده می‌شود:

```
$("#demo").hide();
```

و برای نمایان کردن عناصر از متود `show` استفاده می‌شود:

```
$("#demo").show();
```

می‌توان هر دو کار مخفی و نمایان کردن را با متود `toggle` انجام داد:

```
$("#demo").toggle();
```

اگر می‌خواهید مخفی/نمایان کردن با افکت `Fade` اعمال شود؛ به جای سه متود بالا به ترتیب از توابع زیر استفاده کنید:

```
$("#demo").fadeOut();  
$("#demo").fadeIn();  
$("#demo").fadeToggle();
```

اگر می‌خواهید مخفی/نمایان کردن با افکت `Slide` اعمال شود، از سه متود زیر استفاده کنید:

```
$("#demo").slideUp();  
$("#demo").slideDown();  
$("#demo").slideToggle();
```

متودهای افزودن محتوا به صفحه

برای افزودن محتوای جدید (متن یا عنصر جدید) به عنصر موجود در صفحه، می‌توان از توابع زیر کمک گرفت:

- افزودن محتوا به ابتدای عنصر موجود:

```
$("#demo").prepend('In the name of God<br>');
```

- افزودن محتوا به انتهای عنصر موجود:

```
$("#demo").append(<br>Good bye');
```

- افزودن محتوا به قبل از عنصر موجود:

```
$("#demo").before('In the name of God<br>');
```

- افزودن محتوا به بعد از عنصر موجود:

```
$("#demo").after(<br>Good bye');
```

متود `remove()` عنصر انتخاب شده را حذف می‌کند و متود `empty()` محتوای داخل عنصر انتخاب شده را تخلیه می‌کند.

متودهای تغییر محتوای یک عنصر

برای تغییر یا set کردن محتوای یک عنصر از سه متود زیر استفاده می شود:

- متود html()

○ این متود معادل innerHTML در جاوا اسکریپت است و برای درج کد html در عنصر استفاده می شود.

```
$("#demo").html('<b>In the name of God</b><br>This is a test.<br>Bye!');
```

- متود text()

○ این متود امکان درج کد HTML را ندارد و فقط می تواند متن به عنصر اضافه کند.

```
$("#demo").text('<b>In the name of God</b><br>This is a test.<br>Bye!');
```

- متود val()

○ این متود فقط برای درج محتوا در فیلدهای فریم قابل استفاده است.

```
$("#username").val('Admin');
```

متودهای خواندن محتوای یک عنصر

برای خواندن محتوای عناصر نیز از همان سه متود بالا استفاده می شود با این تفاوت که داخل پرانتز آنها نباید چیزی نوشته شود:

```
var x = $("#demo").html();  
var x = $("#demo").text();  
var username = $("#username").val();
```

متود خواندن/تغییر صفات HTML یک عنصر

برای خواندن مقدار یا تغییر مقدار صفات HTML یک عنصر، از متود attr استفاده می شود.

مثالی از تغییر صفت href یک تگ a

```
$("#go").attr("href", "http://google.com");
```

مثالی از خواندن صفت href یک تگ a

```
$("#go").attr("href");
```

متود خواندن/تغییر صفات CSS یک عنصر

برای خواندن یا تغییر صفات CSS یک عنصر، از متود CSS استفاده می شود.

متود CSS به سه شکل قابل استفاده است:

1 - خواندن مقدار یک صفت:

```
$("#demo").css("background-color");
```

2 - تغییر یک صفت:

```
$("#demo").css("background-color", "yellow");
```

3 - تغییر چند صفت:

```
$("#demo").css({
  "background-color": "yellow",
  "width": "200px",
  "font-family": "Tahoma"
});
```

متود افزودن کلاس به عنصر

برای افزودن یک کلاس CSS خاص به یک عنصر از متود `addClass` استفاده می شود:

```
$("#demo").addClass("test");
```

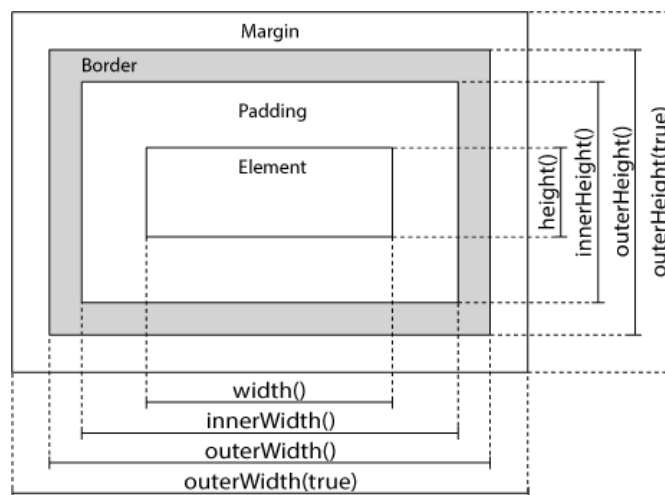
برای حذف یک کلاس خاص از متود `removeClass` استفاده می شود:

```
$("#demo").removeClass("test");
```

متود `toggleClass` بررسی می کند که اگر کلاس مورد نظر در عنصر انتخاب شده وجود داشت، آنرا حذف می کند و اگر وجود نداشت، آنرا اضافه می کند:

```
$("#demo").toggleClass("test");
```

متودهای خواندن/تغییر طول و ارتفاع عنصر



برای خواندن طول یک عنصر چهار حالت (سه متود) وجود دارد که با توجه به تصویر بالا متوجه می شوید هر کدام چه مقداری را برمی گرداند:

```
$("#demo").width();
```

```

$("#demo").innerWidth();
$("#demo").outerWidth();
$("#demo").outerWidth(true);

```

در متودهای بالا اگر یک عدد به عنوان پارامتر ورودی بیان شود، طول عنصر ست می شود؛ مثال:

```

$("#demo").width(500);

```

متود animate

این متود برای متحرک کردن عناصر به کار می رود. یک مثال از نحوه به کار گیری آن:

```

$("#demo").animate({
  width: "300px",
  marginLeft: '30px',
  opacity: '0.5'
});

```

دو نکته:

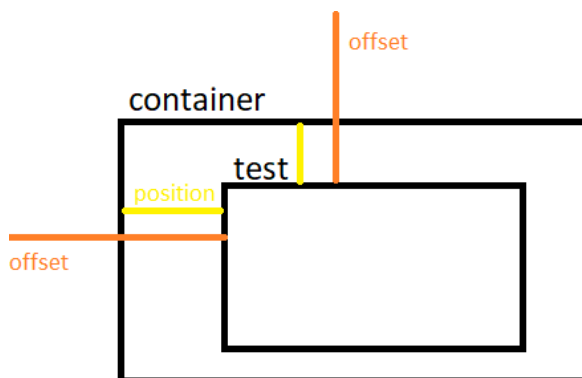
- اطراف نام صفات نیازی به کوتیشن نیست.
- صفاتی که نام دویخشی دارند باید به صورت کوهان شتری نوشته شوند:

○ غلط: margin-left

○ صحیح: marginLeft

توجه: این متود توانایی کار با رنگ ها را ندارد. در صورت نیاز به تغییر تدریجی رنگ، باید از کتابخانه jQuery UI استفاده نمایید.

متود position و offset



متود position() فاصله عنصر از بالا و یا از چپ والد را برمی گرداند.

متود offset() فاصله عنصر از بالا و یا از چپ سند (یا می توان گفته پنجره مرورگر) را برمی گرداند.

```
var xy = $("#demo").offset();  
alert(xy.top);  
alert(xy.left);
```

jQuery UI

این پلاگن این، امکانات گرافیکی مختلفی به جی کوئری اضافه می کند. لیست این امکانات عبارتند از:

Interactions

- Draggable
- Droppable
- Resizable
- Selectable
- Sortable

Widgets

- Accordion
- Autocomplete
- Button
- Checkboxradio
- Controlgroup
- Datepicker
- Dialog
- Menu
- Progressbar
- Selectmenu
- Slider
- Spinner
- Tabs
- Tooltip

Effects

- Add Class
- Color Animation
- Easing
- Effect
- Hide
- Remove Class
- Show
- Switch Class
- Toggle

Toggle Class

Utilities

Position

Widget Factory

jQuery UI را از سایت رسمی آن (<https://jqueryui.com>) دانلود کنید و با کمک کدهای زیر تمامی

فایل‌های آن را به صفحه معرفی کنید:

```
<link rel="stylesheet" href="js/jquery-ui.min.css">
<link rel="stylesheet" href="js/jquery-ui.structure.min.css">
<link rel="stylesheet" href="js/jquery-ui.theme.min.css">
<script src="js/jquery.min.js"></script>
<script src="js/jquery-ui.min.js"></script>
```

به طور مثال با فراخوانی متود `draggable` روی یک عنصر، آن عنصر قابل جابه‌جایی (کشیدن و رها کردن)

می‌شود:

```
$("#demo").draggable();
```

jQuery Mobile

این کتابخانه امکان کار با رخدادهای ویژه موبایل را در برنامه شما فراهم می‌کند.

لیست رخدادهایی که پس از اضافه کردن این کتابخانه به آن‌ها در موبایل دسترسی خواهید داشت عبارتند از:

mobileinit

Event indicating that jQuery Mobile has finished loading.

navigate

A wrapper event for popstate

orientationchange event

Device portrait/landscape orientation event

pagebeforechange

Triggered twice during the page change cycle: First prior to any page loading or transition and next after page loading completes successfully, but before the browser history has been modified by the navigation process.

pagebeforecreate

Triggered on the page being initialized, before most plugin auto-initialization occurs.

pagebeforehide

Triggered on the “fromPage” we are transitioning away from, before the actual transition animation is kicked off.

pagebeforeload

Triggered before any load request is made.

pagebeforeshow

Triggered on the “toPage” we are transitioning to, before the actual transition animation is kicked off.

pagechange

This event is triggered after the changePage() request has finished loading the page into the DOM and all page transition animations have completed.

pagechangefailed

Triggered when the changePage() request fails to load the page.

pagecreate

Triggered when the page has been created in the DOM (via ajax or other) and after all widgets have had an opportunity to enhance the contained markup.

pagehide

Triggered on the “fromPage” after the transition animation has completed.

pageinit

Triggered on the page being initialized, after initialization occurs.

pageload

Triggered after the page is successfully loaded and inserted into the DOM.

pageloadfailed

Triggered if the page load request failed.

pageremove

Triggered just before the framework attempts to remove an external page from the DOM.

pageshow

Triggered on the “toPage” after the transition animation has completed.

scrollstart

Triggers when a scroll begins.

scrollstop

Triggers when a scroll finishes.

swipe

Triggered when a horizontal drag of 30px or more (and less than 30px vertically) occurs within 1 second duration.

swipeleft

Triggered when a swipe event occurs moving in the left direction.

swiperight

Triggered when a swipe event occurs moving in the right direction.

tap

Triggered after a quick, complete touch event.

taphold

Triggered after a sustained complete touch event.

throttledresize

Limits the rate of the execution of handlers on resize events.

updatelayout

Triggered by components within the framework that dynamically show/hide content.

vclick

Virtualized click event handler.

vmousecancel

Virtualized mousecancel event handler.

vmousedown

Virtualized mousedown event handler.

vmousemove

Virtualized mousemove event handler.

vmouseout

Virtualized mouseout event handler.

mouseover

Virtualized mouseover event handler.

mouseup

Virtualized mouseup event handler.

Google Stadia

استی‌دیا: سرویس اشتراکی بازی ابری است که توسط گوگل راه اندازی شده است. این سرویس توانایی پخش آنلاین بازی‌های ویدئویی را تا وضوح ۴کی و با سرعت ۶۰ فریم بر ثانیه دارد. استی‌دیا در دستگاه‌های مختلف از جمله رایانه، تبلت، گوشی همراه و تلویزیون‌ها قابل استفاده خواهد بود.

مطالعه بیشتر:

https://fa.wikipedia.org/wiki/%D%A%AF%D9%88%DA%AF%D9%84_%D8%A7%D8%B3%D8%AA%D8%A7%D8%AF%DB%8C%D8%A7

WebAssembly

WebAssembly نوع جدیدی از کدهای قابل اجرا در مرورگرهای مدرن است که بر خلاف کدهای قدیمی جاوااسکریپت و وی‌بی‌اسکریپت، به صورت کدهای سطح بالا نیست. WebAssembly کدهایی کامپایل شده از یک زبان سطح بالا مانند ++C یا Rust یا Javascript است که با سرعتی بسیار بالا در مرورگرها قابل اجرا است.

WebAssembly به گونه‌ای توسعه یافته است که بتوان از آن در کنار جاوااسکریپت استفاده نمود. توسعه‌ی استاندارد جدید wasm توسط کنسرسیوم وب جهانی و با همکاری شرکت‌های موزیلا، اپل، گوگل و مایکروسافت در حال انجام می‌باشد.

مزیت‌های استفاده از WebAssembly

- سرعت اجرای بسیار بالا در مقایسه با کدهای برنامه‌نویسی قدیمی مانند جاوااسکریپت
- حجم کم کدهای کامپایل شده نسبت به کدهای جاوااسکریپت
- عدم نیاز به کامپایل مجدد در زمان اجرا
- امکان مخفی کردن کدهای سطح بالا توسط برنامه‌نویسان
- امکان برنامه‌نویسی در محیط مرورگر با زبان‌های مختلف و مورد علاقه و در نهایت کامپایل آن

کاربردهای عملی وب اسمبلی

چندین پروژه‌ی جدید در دنیای وب ایجاد شده‌اند که پیش از WebAssembly بسیار دور از ذهن یا گاهی ناممکن بوده‌اند برای مثال کتابخانه‌ی [Opus & Wave Recorder](#) با استفاده از جاوااسکریپت و Web Audio API، صدای کاربر را ضبط می‌کند و در همان محیط مرورگر آن را به OGG تبدیل می‌کند.

کتابخانه‌های دیگری نیز فعالیت‌هایی که پیش از این در محیط مرورگر امکان‌پذیر نبودند را در دسترس قرار داده‌اند.

محدودیت‌های WebAssembly

WebAssembly با هدف ایجاد مزیت سرعت در کنار جاوااسکریپت طراحی شده است و قصد ندارد جایگزین جاوااسکریپت شود. به همین دلیل محدودیت‌هایی نیز دارد که مهم‌ترین آن، عدم دسترسی به DOM است. به عبارت دیگر، کدهای مدیریت و کنترل صفحات وب می‌بایست توسط جاوااسکریپت نوشته شوند و از سرعت و توانایی‌های wasm جهت انجام محاسبات یا پردازش‌های سنگین به ویژه در نرم‌افزارها یا بازی‌های تحت وب کمک گرفته شود.

استفاده از WebAssembly در طراحی سایت

با توجه به اینکه این روش جدید کدنویسی در صفحات وب هنوز به وضعیت پایدار نرسیده است، روشی در دسترس مانند تگ script در html برای آن تعریف نشده است و لازم است تا با استفاده از دستورات جاوااسکریپت، کدهای wasm را ابتدا از هاست دریافت و سپس اجرا نماییم.

[برای اجرای کدهای کامپایل شده در مرورگر می‌توانید این صفحه را مشاهده کنید](#)

پشتیبانی از WebAssembly در مرورگرهای وب

همان‌طور که گفته شده این تکنولوژی هنوز در وضعیت پایدار منتشر نشده است و به همین دلیل اجرای آن در مرورگرها ممکن است با برخی موانع یا ناهماهنگی‌هایی روبرو باشد. ولی به طور کلی اغلب مرورگرهای مدرن از جمله فایرفاکس و کروم، از این تکنولوژی پشتیبانی می‌کنند.

فرمت متنی WebAssembly

مشابه آنچه در زبان اسمبلی وجود دارد، یک زبان سطح پایین با نام WebAssembly Text نیز در حال توسعه است که می‌تواند به صورت مستقیم به WebAssembly کامپایل شود. تولید WebAssembly Text می‌تواند به صورت دستی توسط برنامه‌نویسان یا از تبدیل یک زبان سطح بالا مانند Rust به انجام برسد.