آموزش برنامه نویسی به زبان جاوااسکریپت ، از مقدماتی تا پیشرفته

نام کتاب : جاوااسکریپت سبز ، نسخه 1.1

به نام خداوندی که ذهن مان را روشن نمود

پیشگفتار

درباره ی خودم

فهرست مطالب

وقتی جوان ایرانی بی کار و بی حوصله می بینم شدیدا ناراحت می شوم ، جوانی که تا دیر وقت در فضای مجازی مشغول اتلاف سرمایه جوانی خود هستند قلب من را به درد می آورند ، جوان ایرانی دانه هایی باارزش هستند که امید است در آینده هر یک به درختی تناور و پر شاخه و برگ تبدیل شوند و چندین عابر از میوه ها و سایه ی آن ها بهره ببرند.

همه جوانان ایرانی هر یک در جایگاه خود باارزش و مهم هستند که اگر هر یک از آن ها سستی کند عزت مردم ایران در خطر می افتد.

پسران جوان ایرانی لیاقت دارند که خود صاحب بهترین کسب و کار ها باشند ، رنگ چشم های آن ها ، انگشتان توانمند آن ها همیشه در ذهن و قلب من است ، من به جوانان ایرانی امیدوارم.

دختران ایرانی هر یک به تنهایی همه ی وجود و از گوشت و خون من هستند ، من دوست دارم دختر ایرانی را در حال تلاش برای آخرین رویای آن ها ببینم ، در حال مطالعه ، در حال رشد و در حال زندگی ببینم.

دختر ایران هزاران استعداد و توانمندی دارد، هر یک در زمان خود می توانند چشمه ای جاری باشند و تشنگان علوم مختلف در سرتاسر جهان را سیراب کنند.

دختر ایرانی با هوش و ذکاوت و شوق که دارد می تواند بزرگ ترین شرکت های غول جهان را به چالش بکشد و با آن ها رقابت کند ، من به دختر ایرانی امیدوارم.

دختر ایرانی و پسر ایرانی شایسته این هستند که آرزو کنند و آرزو های آنها به حقیقت تبدیل شوند ، لیاقت دارند که در رویا های خود زندگی کنند.

این کتاب با عشقی فراوان نسبت به جوانان کشورم به خصوص جوانان سردشت نوشته شده است.

امید است که با توفیق خدا و کار جهادی ما ، دانه های ایرانی به درختانی پربار تبدیل شوند.

این کتاب را به یکی از عزیزترین هایم ، رامان عزیزم تقدیم می کنم

با آرزوی پربار شدن دانه های وجودت .

آذر 1400

فصل پنجم : برنامه نویسی شی گرا-------

فصل ششم : DOM & BOM----------

فصل هفتم: فرم و فریم در جاواسکریپت------

بخش اول :

راه و رسم برنامه نویسی

سرعت یادگیری این کتاب نسبت به خودتان تعیین می شود ، همه چیز به شما بستگی دارد ، اگر لازم باشد و یک مبحث برای شما سنگین است می توانید بین مطالعه خود فواصلی را برای استراحت و راه اندازی مجدد مغز خود در نظر بگیرید. تنها راه تبدیل شدن به یک توسعه دهنده موفق این است که به روز باشید و تمرین کنید ، به روز رسانی و تمرین ، به روز رسانی و تمرین.

زبان برنامه نویسی جاواسکریپت

جاوااسکریپت نوعی زبان سطح بالا ، کامپایل درجا (just-in-time) و چند الگویی است. جاوا اسکریپت نحو آکولادی دارد ، نوع دهی آن پویا است ، نوع شی گرایی اش بر پایه پیش نمونه است.و دارای توابع کلاس اول می باشد .

الگوی برنامه نویسی این زبان رویداد محور ، تابعی و دستوری است.
جاواسکریپت یک زبان برنامه نویسی برای وب است. یاد گرفتن این زبان بسیار ساده است. جاوا اسکریپت یکی از سه زبانی است که هر توسعه

دهنده ی وب باید با آن آشنایی داشته باشد . در انتهای این کتاب درباره شئ گرایی در زبان js مفصل توضیح داده شده است.

زبان نشانه گذاری HTML که برای تعریف صفحات وب طراحی شده است

زبان نشانه گذاری CSS که برای تعیین طرح بندی صفحات وب طراحی شده است.

زبان برنامه نویسی جاوااسکریپت که برای برنامه نویسی رفتار صفحات وب طراحی شده است.

در این بخش شما را با برخی از ویژگی های این زبان آشنا می کنم.

Just_in_time کامپایل درجا

جاوااسکریپت در ابتدای شروع کار خود بسیار آهسته بود اما بعدا به لطف JIT بسیار سریع تر شد.

برنامه ها از دو طریق می توانند اجرا شوند :

با استفاده از مفسر

با استفاده از کامپایلر

با استفاده از مفسر عملیات ترجمه خط به خط در زمان اجرای برنامه اتفاق می افتد . در کامپایلر کد ها در حین اجرای برنامه ترجمه نمی شود بلکه برای ترجمه و تبدیل آن به یک فایل اجرایی مدت زمانی را به خود اختصاص می دهد.

مزایا و معایب مفسر

مفسر ها خط به خط جلو می روند و لازم نیست در ابتدا همه چیز را ترجمه کنید و این باعث می شود که نسبت به کامپایلر ها سریع تر باشند.

مرورگر ها در ابتدا از مفسر ها استفاده می کردند زیرا سرعت اجرا برای مرورگر ها خیلی مهم است.

عیب مفسر این است که قطعه کدی که چند بار اجرا می شود باید هر بار ترجمه شود در چنین حالتی زمان اجرا طول می کشد.

مزایا و معایب کامپایلر ها

در ابتدای امر اجرا توسط compiler شما باید چند ثانیه صبر کنید ، زیرا کامپایلر در ابتدا کل برنامه را ترجمه می کند و بعد ها برای اجرا از فایل ترجمه شده دوباره استفاده میکند

عمل ترجمه در ابتدای اجرا باعث می شود که زمان اجرای برنامه افزایش پیدا کند. اما کامپایلر ها یک حسن بزرگ دارند و آن این است که یک کامپایلر کد های بهینه و رفع اشکال شده تحویل می دهد. زیرا با دقت بسیار بیشتری به کد ها نگاه می کند.

توسعه دهندگان فناوری ایجاد کردند که دیگر لازم نیست بین مفسر و کامپایلر یکی را انتخاب کنیم ، بلکه این رویکرد باعث شده که از مزایای هر دو دنیا بهره مند باشیم . JIT ترکیبی از مفسر و کامپایلر است که به دو نوع از ترجمه کد گفته می شود.

JIT مانند یک مفسر عمل می کند و کد در زمان فراخوانی اجرا می شود. با این وجود اگر کدی بار ها و بارها فراخوانی شود برای اجرا مانند کامپایلر عمل می کند.

انواع کد از نظر JIT

Hot code : کد داغ به قطعه کدی گفته می شود که بارها و بار ها فراخوانی شده باشد.

Warm code : این بخش از کد چندین بار فراخوانی شده است اما تعداد فراخوانی شدن آن از کد داغ کمتر است.

Deade code : خط کدی است که هرگز مورد استفاده قرار نگرفته و یا کدی که هرگز فراخوانی نشده است. متغیری که تعریف شده است اما هرگز مورد استفاده قرار نگرفته است یک کد مرده است.

کامپایلر های JUST IN TIME (بهترین های هر دوحالت)

توسعه دهندگان بخش جدیدی را به موتور جاوا اسکریپتی اضافه کرده اند که به آن profile یا مانیتور گفته می شود. این مانیتور در هنگام اجرای کد به آن نگاه می کند و به تعداد مرتبه اجرای کد و نوع هایی که استفاده می کند دقت میکند.

در ابتدا همه کد ها از طریق مفسر اجرا می شوند

به کد ها نشان داغ ، گرم و مرده داده می شود

کامپایلر خط مقدم : وقتی که یک تابع یا یک حلقه داغ باشد JIT آن را برای کامپایل شدن ارسال میکند ، این کامپایل در جایی نگه داری و ذخیره می شود.

Stub : هر خط از کد داغ به واحدی به نام stub کامپایل می شود ، stub ها به صورت خطی توسط یک سری عدد ایندکس گذاری می شوند. اگر مانیتور متوجه شود که این کد با همان نوع داده دوباره اجرا شده نسخه کامپایل شده را برای اجرا می فرستد با این کار همه چیز بسیار سریع می شود.

کامپایلر با مشاهده می تواند متوجه شود که چه راه بهتری برای اجرای کد ما وجود دارد ؟ کامپایلر خط مقدم برخی از بهینه سازی ها را به شرط زمان بر نبودن انجام می دهد.

کامپایلر بهینه ساز : اگر یک کد واقعا داغ باشد حتی با وجود اینکه زمان بر است برای انجام یک سری بهینه سازی ها به کامپیایلر بهینه ساز ارسال می شود.

با انجام عمل بهینه سازی یک نسخه سریع تر از کد داغ ذخیره می شود. مانیتور اطلاعاتی را در اختیار بهینه ساز قرار می دهد که کامپایلر از روی چنین اطلاعاتی مفروضات و قضاوت هایی را به دست می آورد. برای ایجاد یک نسخه سریع تر از کد کامپایلر بهینه ساز از این مفروضات استفاده می کند.

برای مثال اگر کامپایلر متوجه شود که تمام شی های ساخته شده مربوط به یک سازنده یک شکل هستند و شباهت دارند ، بر اساس این فرض یکسری از کد ها را حذف می کند ، البته در خروجی کد تغییری ایجاد نمی شود.

اعتبار سنجی JIT

کد های کامپایل شده نیاز دارند قبل از اجرا شدن درستی فرضیات آن ها بررسی شود. اگر نتیجه این اعتبار سنجی true باشد کد ها اجرا می شوند. در غیر این صورت کد های بهینه سازی شده توسط JIT حذف می شوند. بعد از حذف ، روند اجرا به نسخه ای کامپایلر مقدم تهیه کرده بود و یا به نسخه مفسر باز می گردد . به این جریان deoptimization یا عدم بهینه سازی گفته می شود.

وظیفه مرورگر ها و برنامه های زمان اجرا این است که محدودیت هایی را برای سیکل و deoptimization در نظر بگیرند.

Type spcialization: (تخصصی سازی نوع) یکی از بزرگ ترین کارهایی که کامپایلر های بهینه ساز انجام می دهند . زبان های برنامه نویسی که از سیستم نوع داده ای پویا استفاده میکنند مدت زمان بیشتری را برای اجرا می طلبند. کامپایلر بهینه ساز باعث می شود که JIT در زمان اجرای انواع داده سوال تکراری نپرسد در نتیجه کد بسیار سریع تر اجرا می شود. در کامپیال بهینه ساز تمام تابع با همدیگر کامپایل می شود. بررسی های مربوط به نوع داده ای در این شرایط قبل از اجرای حلقه اجرا شده و دیگر بررسی خاتمه می یابد .

JIT با خود فکر میکند : تکرار مکرر این کارها احمقانه است و باید سعی کنم به طرز هوشمندانه ای این کد را اجرا کنم.

کامپایلر درجا مزیت های هر دو دنیای کامپایلر ها و مفسر ها را در اختیار ما قرار می دهد. می توانیم یک قطعه کد را سریع تر اجرا کنیم که بهینه سازی شده است و از طریق کامپایلر اجرا می شود و همچنین مفسر همیشه آماده است که سایر کد ها را اجرا کند و به دیباگ کردن آسان کمک کند.

انواع کامپایل

کامپایل مبنا : کامپیال مبنا بهینه سازی مبنا را انجام می دهد ، بهینه سازی مبنا یعنی کامپایل کردن کد و سپس ذخیره سازی نتیجه کامپایل بدون صرف

زمان زیاد است . این ساده ترین و سریع ترین کامپایل است. بهینه سازی در این روش ضعیف است . گاهی موجب کاهش سرعت اجرای برنامه می شود.

کامپایل بهینه : این نوع بهینه سازی عمیق تر و طولانی تر است و شامل صرف زمان زیاد در ابتدا و سرمایه گذاری برای بهینه سازی یک بخش از کد با استفاده از کامپایل بهینه و سپس استفاده از مقادیر ذخیره شده آن بهینه سازی می باشد ، به این روش کامپایل optimizing compilation یا opt compiling گفته می شود.

مقایسه کامپایل بهینه و کامپایل مبنا

کامپایل مبنا شبیه این است که شما یک مقاله را فقط با تغییر یک سری علائم نگارشی ، از نظر املا و دستور زبان بهبود ببخشید ولی کامپایل بهینه شبیه به نوعی است که ویرایش محتوایی ، مفهومی و خوانایی یک مقاله انجام گرفته باشد ، در کامپایل بهینه کار زیادی در ابتدا صورت می گیرد اما منجر به نتیجه بهتر می شود.

پیچیدگی زمانی اجرای کدی که با کامپایل شده است ثابت است زیرا ما یک کد بهینه را بارها و بارها اجرا میکنیم

انتخاب JIT

JIT باید انتخاب کند که یک قطعه کد توسط کدام کامپایلر اجرا شود. JIT این تصمیم گیری را با استفاده از برچسپ داغ یا گرم انجام می دهد. کد گرم توسط کامپایلر بهینه سازی و ارجا می شد.

JIT به ما امکان می دهد تنها به چیز هایی سرعت ببخشیم که لازم است سریع باشند.

انواع سطح زبان

این که یک زبان برنامه نویسی در دسته زبان های سطح بالا یا پایین قرار می گیرد به تجرید یا (abstraction) و میزان نزدیکی کارکارد آن به سیستم عامل مربوط است .

ویژگی های زبان سطح بالا :

قابلیت تجرید یا انتزاع داند

به زبان انسان نزدیک تر بوده و خوانایی بهتر دارند.

کاری به مدیریت حافظه ندارد

ویژگی های زبان سطح پایین

قابلیت انتزاع را ندارند

به زبان ماشین نوشته می شوند و درک آن برای انسان سخت است.

نیاز به مدیریت حافظه دارد

نمونه هایی از زبان های برنامه نویسی سطح پایین شامل زبان (اسمبلی) و کد ماشین است

زبان های C و یا C++ بین زبان های سطح بالا و سطح پایین قرار دارند این زبان ها امکان مدیریت حافظه را ارائه می کنند از سوی دیگر قابلیت تجرید نیز دارند.

زبان های سطح پایین Low Level Language

زبان های سطح پایین به سیستم رایانه ای نزدیک هستند. یکی از رایج ترین زبان های سطح پایین کد ماشین است. کد ماشین هیچ تجریدی ندارد و شامل دستورالعمل های منفرد است که به صورت رایانه ارسال می شوند. ماشین ها تنها بایت ها را درک میکنند که به صورت دودویی نمایش می یابند. گرچه ممکن است گاهی با نماد گذاری ده دهی یا شانزدهی نوشته شوند.

کد ماشین سریع تر و سخت ترین زبان برنامه نویسی است.

زبان اسمبلی : یک زبان سطح پایین است که یک گام از بالاتر از زبان کد ماشین قرار دارد. زبان اسمبلی از تجرید اندکی برخوردار است. این زبان شبیه کد ماشین است و نسبت به زبان C کمتر مورد استفاده قرار می گیرد

زبان C: این زبان نزدیک به کد ماشین است . اغلب عملیاتی که در زبان c نوشته می شوند می توانند با چند دستورالعمل صدور کد ماشین کار کنند.

زبان های سطح بالا High Level Language

هر زبان سطح بالا ساختار نگارشی خاص خود را دارد. این زبان نسبت به زبان سطح پایین کند تر است اما یادگرفتن آن نسبت به زبان سطح پایین آسان تر است.

این زبان قابلیت تجرید یا انتزاع دارد و کاملا خوانا و آسان است. نیاز به مدیریت حافظه ندارد و درگیر تخصیص حافظه نمی شوید و به خاطر همین این زبان ها امن هستند و به رایانه صدمه نمی زنند.

نمونه هایی از زبان سطح بالا شامل : پایتون ، جاوا ، سی شارپ ، کاتلین و زبان جاوا اسکریپت است.

زبان های خاص منظوره و همه منظوره

زبان هایی که فقط برای یک هدف ایجاد شده و استفاده می شوند را زبان های برنامه نویسی خاص منظوره یا خاص دامنه (Domain Specific)می گویند.

زبان نشانه گذاری HTML = برای طراحی صفحات وب

زبان SQL= برای کار با پایگاه داده رابطه ای

VHDl= برای توصیف سخت افزار

زبان هایی که برای نوشتن نرم افزار در زمینه های متنوع به کار می روند را زبان های عام منظوره یا همه منظوره (General purpose) می گویند ، نظیر C,PHP ، پایتون و جاوا

تجرید ، انتزاع ، Abstruct

انتزاع ، تجرید یا فرآیند اختصار ، فشرده سازی و تلخیص اطلاعات از طریق شناسایی ، استخراج و سپس جداسازی و پنهان سازی جزئیات از کلیات است . انتزاع در لغت به معنای جدا کردن ، گرفتن ، درآوردن ، جزئی از کل و به معنای بازداشتن و امتناع کردن ، برکندن است.

در این فرآیند کلیات مدنظر بوده و به جزئیات پرداخته نمی شود و چه اندازه از جزئیات را حذف کنیم سطح انتزاع را نشان می دهد.

مثلا وقتی در خیابان از کنار یک ماشین رد می شوید چندین دید می توانید درباره آن ماشین داشته باشید

ماشینی که از کنار آن رد شدم مشکی و گران قیمت بود

ماشینی که از کنار آن رد شدم یک ماشین ، مشکی ، گران قیمت و خارجی بود

ماشینی که از کنار آن رد شدم یک ماشین ، مشکی ، گران قیمت و خارجی بود که چراغ های زیبایی داشت.

انتزاع مفهوم یا ایده ای که وابسته به هیچ نمونه خاصی نباشد است. در علم رایانه انتزاع اطلاعات جزئی را کاهش دادن به طوری که شخص بتواند روی مفاهیم تمرکز کند می باشد. تجرید یا انتزاع به معنای تفکیک مباحث مربوط به هم و نگریستن به موضوع جدای از مباحث وابسته به آن است.

تجرید به مدیریت پیچیدگی و پیشگیری از وابستگی (tight coupling) اجزای سیستم به یکدیگر است که باید در سامانه های نرم افزاری تا حد امکان از آن کاست.

الگوی برنامه نویسی

الگوریتم برنامه نویسی ، پارادایم برنامه نویسی به معماری به کار گرفته شده در کد نویسی ها اشاره دارد و به معنای نوع کد نویسی است

الگوی برنامه نویسی مسیر یا سبک برنامه نویسی است.

انواع الگو های برنامه نویسی

برنامه نویسی شی گرا obgect oriented programming

برنامه نویسی تابعی functional programming

برنامه نویسی رویه ای procedural programming

برنامه نویسی غیر ساخت یافته

برنامه نویسی رویداد گرا Event driven programming

برنامه نویسی پویا Dynamic programming

برنامه نویسی منطقی Logic programming

برنامه نویسی مبتنی بر خودکار سازی Automata _based programming

برنامه نویسی اعلانی

برنامه نویسی دستوری

زبان های برنامه نویسی چند الگویی: زبان های برنامه نویسی این قابلیت را دارند که از یک یا چند شیوه برنامه نویسی پشتیبانی کنند برای مثال برنامه های نوشته شده با سی پلاس پلاس می توانند به طور کامل رویه ای یا منطبق بر شیوه برنامه نویسی شی گرا باشند . یا حتی می تواند حاوی عناصری از هر دو شیوه باشد.

زبان های برنامه نویسی شی گرا

PHP , Python ,C#,ruby , Perl، سوئیفت

زبان های برنامه نویسی دستوری

js ,PHP , Python ,C#,ruby , Perl

زبان های برنامه نویسی تابعی

PHP , Python ,C#,ruby , Perl، سوئیفت ,Lisp, js

زبان های برنامه نویسی رویه ای

PHP ، Lisp، Perl

زبان های برنامه نویسی رویداد محور

Js,C#, Perl

زبان های برنامه نویسی ساخت یافته

C#

زبان های برنامه نویسی Dynamic & Static

زبان های برنامه نویسی از لحاظ نوع (type) به دو دسته dynamicType و StaticType تقسیم می شوند. در زمان اجرای برنامه (run_time) یا در زمان compile برخی اعتبار سنجی ها روی برنامه صورت می گیرد که اساس تفاوت زبان های Static و Dynamic را مشخص میکند.

زبان برنامه نویسی با نوع دهی Dynamic

در این زبان هنگام کد نویسی نیاز نیست که نوع متغیر را مشخص کنید در این زبان نیازی به اعلان (declaration explicit) متغیر قبل از استفاده نیست.

بیشتر زبان های اسکریپتی در این دسته قرار می گیرند.

نوع متغیر درزمان اجرا (run-Time) و با توجه به مقدار متغیر مشخص می شود.

Interpreter ها معمولا در هنگام اجرا TypeChecking را انجام می دهند.

برخی از زبان هایDynamic Type عبارتند از : , js ,PHP,Python Ruby , Perl

زبان برنامه نویسی با نوع دهی Static

به زبانی که قبل از compile با نوع متغیر ها مشخص باشد زبان های Static Type نامیده می شود. مهم ترین ویژگی استاتیک بودن یک زبان این است که بخش اعظم خطاها توسط کامپایلر برطرف شده و در هنگام اجرای کد احتمالا با خطای کمتری مواجه خواهیم بود. در این زبان TypeChecking قبل از اجرای کد و در هنگام compile انجام می شود.

برخی از زبان های Static Type عبارتند از : Go, C, Java , C# , C++

اسکریپت چیست؟ زبان برنامه نویسی اسکریپتی چیست؟

اسکریپت یک برنامه یا دستور العمل های نوشته شده با استفاده از یک زبان اسکریپتی است. اسکریپت به کد های برنامه نویسی گفته می شودکه کامپایل نمی شوند و معمولا تفسیر می شوند.

اسکریپت ها برناه های کوچکی هستند که با استفاده از آن ها می توان وظایف طولانی و تکراری را به صورت خودکار انجام داد.

زبان های اسکریپتی (مفسری) زبان هایی هستند که زمان ترجمه و زمان اجرا یکی است و از هم جدا نیستند . در واقع عمل ترجمه و اجرا در این زبان ها همزمان انجام میشود.

برای مثال در زبان c++ وقتی گزینه قعد را می زنید ابتدا کد ترجمه می شود ، سپس اگر کد بدون خطا بود اجرا می شود. ولی در یک زبان اسکریپتی وقتی گزینه run را می زنید عمل ترجمه و اجرا به صورت همزمان از همان دستور اول شروع می شود و هر وقت به یک خطا رسید متوقف می شود.

زبان های اسکریپتی مثل python , js ,perl

اکما اسکریپت چیست؟

اکما اسکریپت Ecma Script یا ES

مشخصات یک زبان اسکریپتی استاندارد شده است ، اکما اسکریپت برای استاندارد سازی جاوا اسکریپت ایجاد شده است تا بتواند پیاده سازی جداگانه مستقل را تقویت کند.

زبان اکما اسکریپت شامل ویژگی های ساختاری ، سینتکس ساده ، پویا ، برنامه نویسی تابعی و شی گرایی مبتنی بر پیش نمونه است.

زبان Jscript و javaScript از زبان های اکما اسکریپت هستند. در واقع این دو ، دو نام متفاوت برای یک زبان هستند. دلیل این تفاوت دور زدن مسائل مربوط به علامت تجاری بود. دستور هایی که به Jscript اضافه شده اند به صورت ویژه ای با مرورگر Internet Explorerمایکروسافت هماهنگی دارد.

کاربرد زبان جاوااسکریپت در کدام حوزه ها می باشد؟

زبان جاوااسکریپت یک زبان همه فن حریف است از این زبان می توان برای برنامه نویسی سمت سرور ، اپلیکیشن های بازی ، اپلیکیشن های دسک تاپ استفاده کرد . به صورت کالی می توان کاربرد های زبان جاوااسکریپت را به صورت زیر بیان کرد:

برنامه نویسی فرانت اند

برنامه نویسی بک اند

برنامه نویسی نرم افزار های موبایل

برنامه نویسی نرم افزار های دسک تاپ

با استفاده از جاوااسکریپت می توانید وب سایت های پویا تر و جذاب تری برای کاربران خود بسازید . فریم ورک ها و ابزار های بسیار زیادی دارد که با استفاده از آن ها می توانید در اکثر حوزه ها برنامه نویسی کنید ، اکثر مرورگر ها از زبان جاوااسکریپت پشتیبانی می کنند . تشخیص خطا و دیباگ در جاوااسکریپت مشکل است .

به چند روش می توان از کد های js استفاده کرد؟

زبان برنامه نویسی جاوااسکریپت در سمت سرور و سمت سرویس گیرنده یا کاربر استفاده می شود ، در سمت کاربر همراه با زبان های HTML , CSS استفاده می شود.

در سمت کاربر از زبان برنامه نویسی جاواسکریپت به دو روش می توان استفاده کرد :

استفاده در داخل کد های HTML

استفاده به صورت یک فایل خارجی

کدهای جاوااسکریپت در صفحه HTML باید بین تگ های باز و بسته <script> </script> قرار داده شود.

کد های جاوااسکریپت را می توان در صفحه HTML بین تگ های زیر قرار داد:

<body></body>

<head></head>

می توان به صورت نامحدود هر تعداد خط اسکریپت را در صفحه HTML قرار داد و همچنین می توانید اسکریپت ها را همزمان در دو قسمت <body> و <head> بیاورید. اما بهتر است برای جلوگیری از تداخل با محتویات صفحه تمام توابع را در قسمت <head> صفحه و یا در پایین صفحه قرار دهید.

توابع در بین تگ های باز و بسته <script> </script> تعریف می شوند اما ممکن است که خارج از این تگ ها فراخوانی شوند.

در مثال زیر یک قطعه کد به زبان js در میان تگ های باز و بسته <script> </script> قرار دارد .این یک مثال است برای این که شما متوجه شوید که چگونه کد های جاوااسکریپت در میان کد های HTML قرار می گیرند.

```
<html>
<head>

<body>
```

```
    <script>
        var name=prompt("whats your name?"
);
        var family=prompt("what is your fa
mily"));

        console.log(name+family);
    </script>
</body>
</html>
```

برای مشاهده خروجی بعد از دو بار کلیک کردن بر روی فایل با فرمت html و اجرا شدن آن توسط مرورگر ، بر روی صفحه مرورگر راست کرده و برروی گزینه inspect کلیک کنید.

و یا بعد از اجرا شدن فایل تو سط مرورگر ، دکمه های ترکیبی ctrl + shift + c را فشار دهید.

استفاده از یک فایل Javascript خارجی

این فایل ها با فرمت js. ذخیره می شوند و کد های جاواسکریپت در آن ذخیره می شود. اسکریپت خارجی نمی تواند شامل تگ های باز و بسته <script> </script> باشد.

در مثال زیر کد های زبان js در فایلی به نام script.js ذخیره شده اند و به صورت زیر می توان از دستورات این فایل خارجی استفاده کرد.

```
<html>
<head>
<script  type="text/javascript" src="scrip
t.js"></script>
</head>
<body>

</body>
</html>
```

فایل های خارجی جاواسکریپت اغلب شامل کد هایی است که در چندین صفحه HTML مختلف استفاده می شود.

اگر کد های جاواسکریپت شما خیلی طولانی باشند بهتر است در یک فایل خارجی قرار بگیرد. همچنین توصیه می شود که کد های جاواسکریپت در قسمت body یک صفحه HTML قرار بگیرند زیرا head یک صفحه برای موتور های جستجو و از نظر سئو باید تمیز باشد و باعث می شود سرعت بارگذاری سایت بالا رود.

مطالبی که بیان شد برای زمتنی استفاده میشود که شما زبان جاوسکریپت را همراه با زبان های HTml و CSS به کار می برید. در غیر این صورت زمانی که زبان جاواسکریپت در سمت سرور توسط Nodejsاجرا می شود اصلا نیازی به تگ باز و بسته <script> </script> ندارد و با فرمت js. ذخیره و خارج از مرورگر اجرا می شود.

بهترین روش برای یاد گرفتن یک زبان برنامه نویسی چیست؟

روش های متفاوتی برایادگرفتن یک زبان برنامه نویسی وجود دارد و بهترین روش ، روشی است که برای شما سودمند و لذت بخش باشد.

یادگرفتن از طریق کتاب های آموزش برنامه نویسی

یک کتاب می تواند از نظر اطلاعات کافی و کامل باشد اما نیازی نیست که شما در ابتدا همه چیز را حفظ کنید و دانش شما به مرور زمان با انجام دادن پروژه های مختلف تکمیل می شود.

اما در کل کتاب ها مرجع خوبی برای مواقع ضروری به حساب می آیند .

یادگرفتن از طریق گوگل و جستجو در اینترنت

در این روش یاد گرفتن خیلی سریع ، ساده و شیرین می شود و اینترنت همیشه بزرگ ترین منبع و استاد یک برنامه نویس است اما بهتر است شما برای یاد گرفتن خود یک برنامه یا فهرست از قبل تهیه کنید تا در میان حجم انبوه اطلاعات گوگل گم نشوید.

یادگرفتن از طریق کارآموزی در تیم های برنامه نویسی

اگر از این روش آموختن بهره مند باشید واقعا تجربه باارزشی است ، چون بهترین روش یادگرفتن بودن در کنار افراد با تجربه است و شاید بتوانید در چند ماه یا چند روز از تجربه چند ساله یک فرد متخصص استفاده کنید و بهره مند شوید ، اما بهتر است بدانید که همه تیم ها و اساتید به خوبی با کنجکاوی یک فرد مبتدی برخورد نمی کنند و شاید باعث شوند که علاقه یا شور کارآموز برای یادگرفتن از بین برود.

یادگرفتن از طریق رفتن به کلاس های برنامه نویسی

این روش از بقیه روش ها گران تر است و مزایا و معایب خود را دارد اما در کل همه چیز به دانش و نحوه تدریس استاد وابسته است.

یادگرفتن از طریق تماشای فیلم های آموزشی

در این روش مطالب اصلی به زبان ساده و صمیمی بیان می شود ، معمولا پروژه محور هستند ، برخی از این فیلم و دوره ها با پشتیبانی مدرس ارائه می شوند. اغلب در این روش ابزار های مورد نیاز شما هم آموزش داده می شود.

من شخصا همه روش های نام برده را استفاده کرده ام و همه روش ها مفید بوده اند اما اصل یاد گرفتن با کتاب و تمرین فراوان ممکن می شود . اگر مطالب اصلی را به واسطه کتاب عمقی یاد بگیریدو روزانه تمرین مداوم داشته باشید حتما برایتان مفید خواهد بود.

Nodejs چیست؟

Nodejsیک محیط اجرا برای کد های جاوااسکریپت است ، به واسطه Nodejs می توانیم از زبان js برای سمت سرور استفاده کنیم . Nodejs یک مفسر و محیط برای اجرای زبان جاوااسکریپت است. Nodejs یک زبان برنامه نویسی نیست. این ابزار معمولا توسط برنامه نویسی که از زبان جاواسکریپت برای سمت سرور استفاده می کند استفاده می شود.

موتو جاوااسکریپت یک مفسر کامپایل درجا است که کد های به زبان جاواسکریپت را ترجمه و اجرا میکند . موتور جاواسکریپتی در مرورگر قرار دارد. برای همین در گذشته زبان جاواسکریپت فقط توسط مرورگرها اجرا می شد.

اما در سال 2009 Nodejs توسط رایان دل نوشته شد. نود جی اس از موتور جاواسکریپتی V8 گوگل به همراه یک حلقه رویداد و یک رابط برنامه نویسی کاربردی سطح پایین برای ورودی و خروجی ارائه شد.

این ابزار به زبان javascript , C, C++ نوشته شده است.

کتابخانه های زبان جاوااسکریپت

کتابخانه یک قطعه کد است که شما می توانید بارها از ان استفاده کنید ، یک کتابخانه می تواند شامل چنیدن متد ، توابع و اشیاء وابسته به زبان برنامه نویسی هدف باشد . برنامه ای که که می نویسیم برای استفاده از کتابخانه باید به آن لینک شود.

برخی از کتابخانه ای زبان جاوااسکریپت

D3.js

jQuery.js

chart.js

React.js

Glimmer.js

Bideo.js

Micron.js

Socket.Io

فریمورک های زبان جاوااسکریپت

فریمورک ها شامل چنیدن کتابخانه هستند ، یک فریم ورک برنامه ای عظیم از پیش نوشته شده برای حل یک مشکل خاص است. فریمورک یا چهارچوب کنترل بیشتری روی برنامه شما دارد.

زمانی که از یک فریم ورک استفاده میکنیم چهارچوب و اسکلت بندی پروژه بر پایه آن فریمورک بنا می شود. از چندین فریمورک نمی توان به صورت همزمان استفاده کرد ، اما می توان از چنیدن کتابخانه به صورت همزمان استفاده کرد.

فریمورک های فرانت اند

Angular.js

Vue.js

Aurelia.js

Ember.js

Gatsby.js

Nuxt.js

Next.js

Mithril.js

Backbone.js

Gridsome

فریمورک های بک اند

Express.js

Meteor

فریمورک های موبایل و دسک تاپ

React Native

Electron.js

نکاتی درباره برنامه نویسی به زبان جاوااسکریپت

جاوااسکریپت نسبت به کوچکی و بزرگی حروف حساس است. برای مثال دو متغیر num و Num دو متغیر متفاوت محسوب می شوند.

گذاشتن ; (سمی کولن) در انتهای دستورات جاوااسکریپت اختیاری است ، اما با گذاشتن سمی کولن در انتهای هر دستور باعث می شوید که برنامه خوانا تر شود

توضیحات در جاوااسکریپت

در همه زبان های برنامه نویسی توضیحات به صورت یک خطی و چند خطی وجود دارد. اگر بخواهید یک بخشی از قطعه کد اجرا نشود و توسط مفسر نادیده گرفته شود آن را به صورت توضیحات و یا کامنت به برنامه خود اضافه می کنید.

کدی که به صورت توضیحات به برنامه اضافه می شود توسط کابر دیده نمی شود و هیچ عمل خاصی را جز برای بالا بردن خوانایی برنامه انجام نمی دهد.

در زبان جاوااسکریپت توضیحات تک خطی بعد از // قرار می گیرد.

و توضیحات چند خطی بین /* و */ قرار می گیرد.

```
// this is single line comment
/*
this is multi
line comment
*/
```

انواع زبان ها

در رشته های ریاضی و رایانه زبان ها به دو دسته زبان های صوری (Formal languages)و زبان های طبیعی(Natural languages) تقسیم می شوند.

زبان های صوری یا فرمال زبان های هستند که برای کاربرد های خاص و بیان های خاص از قبیل نمایش تئوری های ریاضی و یا برنامه های کامپیوتری توسط انسان ها تعریف شده اند. زبان‌های فرمال زبان‌هایی هستند که توسّط گرامرها تولید می‌شوند یا ماشینی برای ارزیابی آن‌ها وجود دارد. همه زبان های برنامه نویسی زبان صوری ویا زبان رسمی هستند.

زبان های طبیعی زبان هایی هستند که مردم با آن صحبت می کنند و توسط هیچ کس خاصی طراحی نشده است ، بلک هاین زبان ها در طی زمان ساخته شده اند. زبان های طبیعی مثل زبان فارسی ، کردی ، انگلیسی و ...

زبان های قوانین نوشتن خاصی دارند ، همان طور که هر زبان طبیعی قانون خاص و طبیعی خودش را دارد.

برای مثال زبان انگلیسی قوانین نوشتن مخصوص به خود را دارد و از طرف دیگر قوانینی دارد که در همه زبان های طبیعی یکسان است.

برای مثال درریاضیات3+3=6 یک عبارت است ، که نحوه نگارش صحیح آن به این شکل است و برای مثال به صورت 3$=3+6 صحیح نیست. قوانین نحوه نگارش زبان های رسمی دو نوع هستند : قوانین مربوط به توکن ها و قوانین مربوط به ساختار عبارات.

توکن ها عناصر اولیه یک زبان هستند ، و هر زبانی توکن های خاص خودش را دارد ، مثل اعداد ، حروف ، علائم ، عناصر شیمیایی و..

قانون دوم قوانین نحوی مربوط به ساختار عبارت است. ساختار یک عبارت در واقع همان سبک آرایش و چیدمان توکن ها می باشد ، باید چیدمان توکن های مجاز یک زبان بر اساس ساختار درست آن باشد .
برای نگارش درست یک عبارت در یک زبان صوری باید از توکن های مجاز به شکلی مجاز استفاده کنیم.

به جمله های زیر توجه کنید:

بیرون کریم آمد.

کذیم $bیرون @مد.

جمله اول نادرست است ، توکن های آن مجاز هستند اما ساختار آن صحیح نیست.

جمله دوم هم نادرست است چون از توکن های مجاز استفاده نشده است ، اما ساختار آن از نظر چیدمان توکن ها صحصح می باشد.

وقتی یک جمله به زبان طبیعی را میخوانید و یا می شنوید به صورت ناخوآگاه آن را تحلیل میکنید و درک می کنید که این جمله چه معنای دارد و از بخش هایی مثل فعل ، نهاد ، فاعل یا مفعول تشکیل شده است.
تجزیه زبان های طبیعی به صورت ناخودآگاه انجام می شود.

تجزیه یعنی بررسی ساختار نحوی یک زبان و درک معنا و کاکرد آن .

انسان می تواند یک یا چند عبارت نوشته شده با یک زبان صوری را بخواند و تجزیه کند ، این کار را با شناخت درست از توکن ها و ساختار آن زبان انجام می دهد.

تفاوت های زبان های طبیعی و صوری

ابهام: در یک زبان طبیعی ابهامات زیادی وجود که مردم توسط یک سری نشانه یا اطلاعات قبلی می توانند معنای این ابهامات را درک کنند ولی زبان های صوری طوری طراحی شده اند که خالی از ابهام باشند. هر عبارت به زبان صوری صرف نظر از مفهوم فقط یک معنی می دهد

افزونگی : در یک زبان طبیعی به منظور رفع ابهامات از افزونگی استفاده می کنند ، اما یک زبان رسمی افزونگی ندارد.

لفظ : یک زبان طبیعی کنایه ها و اصطلاحات بی شماری دارد و گاهی یک جمله اصلا معنی واقعی خودش را نمی دهد بلکه به یک معنی دیگر برداشت می شود، اما در یک زبان رسمی هر عبارت فقط یک معنی می دهد.

وقتی یک متن به زبان طبیعی را می خوانیم آن را از سمت راست یا چپ و از بالا شروع به خواندن می کنیم ، اما در یک زبان صوری شما باید متن یا کد را تجزیه کنید ، ساختاررا تفسیر کنید و نشانه ها تشخیص دهید.

برنامه چیست؟ یک یا چند الگوریتم برای حل یک مسئله است و به یکی از زبان های برنامه نویسی نوشته است. یک برنامه خالی از ابهام و افزونگی می باشد و توسط تجزیه و تحلیل ساختار و توکن ها درک می شود و توسط کامپیوتر ها اجرا می شود.

اشکال زدایی در برنامه نویسی

انسان موجودی است که در هر کاری می تواند خطا داشته باشد و برنامه ها چون توسط انسان ها نوشته می شوند از خطا برخوردار هستند. به خطا های برنامه نویسی Bug
و یا Error گفته می شود . و به عمل تشخیص ، جداسازی و تصحیح آن خطاها Debuging گفته می شود.

در یک برنامه ممکن است با سه نوع خطا مواجه شوید :

خطا های نحوی(syntax error)

این نوع خطا ساده ترین نوع از خطای های یک برنامه است ، باید از قوانین نحوی مثل قوانین توکن های مجاز یک زبان و ساختار مجاز یک زبان پیروی کنید تا برنامه شما قابل تجزیه باشد. اگر برنامه شما حتی فقط یک خطای نحوی داشته باشد تجزیه آن غیر ممکن می شود. زمانیکه در برنامه نویسی تازه کار هستید زمان زیادی را صرف برطرف کردن خطا های نحوی خواهید کرد اما نترسید شما به مرور زمان خیلی کمتر با این خطاها مواجه خواهید شد و سریعتر می توانید آن ها را تشخیص دهید.

خطاهای زمان اجرا(runtime error)

این خطا ها در زمان اجرای یک برنامه رخ می دهد و از ادامه اجرای آن جلوگیری میکند .تا زمانی که برنامه اجرا نشود شما از وجود این خطا ها بی خبر هستید . به این نوع خطا ها اعتراض هم گفته می شود ، زیرا به وجود یک اتفاق بد اعتراض میکند.

خطا های معنایی (semantic error)

زمانی که یک خطای معنایی رخ می دهد شما هیچ پیغام خطا و یا اعتراضی را مشاهده نخواهید کرد و برنامه با موفقیت اجرا می شود . اما نتیجه برنامه درست نیست.

شما باید برگردید عقب کد را بخوانید و تجزیه کنید تا بفهمید در کجای برنامه شما منظور خود را درست بیان نکرده اید . برطرف کردن این نوع خطا نیاز به مهارت و به نسبت انواع دیگر خطا ها سخت تر می باشد.

اشکال زدایی

شما به مرور زمان اشکال زدایی را یاد می گیرید ، شما یاد می گیرید یک گام جلوتر از مفسر عمل کنید و هر خط کدی که می نویسید را از قبل در ذهن خود تجزیه کنید ، سپس آن را در ویرایشگر کد می نویسید و اجرا میکنید و به یک خطا بر می خورید و یا به نتیجه دلخواه خود نمی رسید.

خطا یک سرنخ به شما می دهد و شما به یک فرضیه می رسید که اگر تغییری ایجاد کنید کد درست عمل خواهد کرد و سپس فرضیه را عملی میکنید . فرضیه شما ممکن است صحیح یا غلط باشد. شما فرضیه های دیگری هم به کار خواهید گرفت و این روند سعی و خطا ادامه می یابد تا زمانی که شما به یک برنامه درست و کامل می رسید.

دستور چاپ

ساده ترین و پرکاربرد ترین دستور در هر زبانی دستور چاپ خروجی می باشد.

برای چاپ خروجی در زبان جاوااسکریپت از console.log () استفاده می شود.

Console یک ماژول مربوط به محیط های اجرا مثل مرورگر و nodejs می باشد و متد های مختلفی دارد که متد های آن را در ادامه این فصل کامل بررسی خواهیم کرد.

می توان هر چیزی را با دستور چاپ در خروجی چاپ کرد.

```
<script>
   console.log("hello world!");
   console.log("my name is samira abdi");
   var pi=3.14;
   console.log(pi);
   var name="raman";
   var family="eshghie";
   console.log(name+family);
   console.log(2+2);
   console.log(true);

</script>
```

با استفاده از دستور اول و دوم عبارت های hello world! و my name is samira abdi در خروجی چاپ می شوند. با استفاده از دستور سوم مقدار 3.14 در متغیری به نام pi ذخیره می شود. دستور چهارم مقدار داخل متغیر pi را چاپ می کند. دستور پنجم و ششم به

ترتیب دو متغیر تعریف کرده اند برای مقدار نام و فامیلی و در دستور هفتم این دو مقدار الحاق می شوند و چاپ می شوند . دستور هشتم حاصل 2+2 را چاپ می کند و دستور نهم مقدار trueرا چاپ میکند.

خروجی

hello world!

my name is samira abdi

3.14

ramaneshghie

4

true

متغیر ها

مقادیری که در یک برنامه ما با آن ها سروکار داریم ، در خانه ای حافظه ذخیره می شوند ، این خانه ها مثل یک ظرف هستند برای نگهداری ورودی ها ، خروجی ها و مقادیری که برنامه روی آن محاسباتی انجام می دهد. برای دسترسی به این خانه ها و استفاده آسان تر از آن ها ما به هر یک از این خانه ها یک نام اختصاص می دهیم ، این نام باید منحصر بفرد باشد .پس متغیر نامی است که به یک مقدار اشاره می کند.نام متغیر با آدرس آن تفاوت دارد، آدرس ذخیره سازی یک مقدار توسط برنامه نویس

مشخص نمی شود اما نام متغیر توسط برنامه نویس انتخاب می شود. مقادیری که داخل متغیر ها ذخیره می شوند می توانند در طول اجرای برنامه تغییر کنند برای همین به این مقادیر لقب متغیر داده شده است.متغیر نامی است برای خان های حافظه که مقادیر محاسباتی یک برنامه را ذخیره می کنند.

نوع یک متغیر با توجه به نوع مقداری که در آن ذخیره می شود مشخص می شود. یک متغیر می تواند تعریف شود و سپس مقداری در آن ذخیره شود. یک متغیر می تواند مقدار هیچ یا Null را بپذیرد.

یک متغیر به سه روش می تواند مقدار دریافت کند:

دریافت مقدار از ورودی توسط کاربر: برای مثال می توانیم نام و نام خانوادگی کاربر را بپرسیم و در دو متغیر ذخیره کنیم.

تخصیص مقدار توسط برنامه نویس: برنامه نویس مقدار عدد پی را از قبل می داند ، پس مقدار 3.14 توسط برنامه نویس در متغیر pi ذخیره می شود.

نتیجه یک محاسبه : می توانیم با استفاده از محاسبات نتیجه یک عبارت مثل مساحت دایره را بدست آوریم و در یک متغیر مثل area ذخیره کنیم.

زبان جاوااسکریپت یک زبان نوع دهی پویا است و نیازی نیست که کاربر در هنگام تعریف متغیر نوع مقداری که در ان ذخیره می شود را تعیین کند.

کلمات کلیدی و اسامی متغیر ها

تعیین نام منحصر به فرد برای خانه های حافظه قوانینی دارد:

برای نامگذاری متغیر ها می توانیم از حروف A-Z و a-z استفاده کنیم.

استفاده از اعداد در نامگذاری متغیر مجاز است اما نمی توانیم در ابتدای نام متغیر از عدد استفاده کنیم.

استفاده از علائم و نشانه ها مثل (!@#$%^&*() به غیر از underline در نام گذاری متغیر ها غیر مجاز است.

استفاده از space در نامگذاری متغیر ها غیر مجاز است.

استفاده از کلمات کلیدی یک زبان برای نامگذاری متغیر ها غیر مجاز است. کلمات کلیدی مثل for , if
while,else,class,object,list,array, و... کلماتی هستند که از قبل برای کاربرد های خاصی رزرو شده اند . اگر از ویرایشگر هایی مثل VS Code استفاده کنید ، رنگ کلمات کلیدی با رنگ بقیه کد ها متفاوت است.

تعریف متغیر ها در زبان جاواسکریپت

در زبان جاواسکریپت متغیر ها با استفاده از کلمات کلیدی مثل var , let , const تعریف می شوند.

```
var name="raman";
```

```
var family="eshghie";
```

با استفاده از کد بالا دو متغیر به نام های name , family تعریف می شوند و در حین تعریف مقدار دهی هم می شوند.

```
var first_name;
first_name="samira";
```

در کد بالا ابتدا متغیر در دستور اول تعریف شده و سپس به آن مقدار دهی شده است.

```
let area;
let length=4;
let width=3;
area=length*width;
console.log("area=",area);
```

در کد بالا سه متغیر تعریف شدند به نام های length,width,area تعریف شده اند. به length مقدار 4 و به widthمقدار 3 اختنصاص داده شده است. بعد از محاسبه مساحت مقدار آن در area ذخیره می شود. و سپس با استفاده از دستور پنجم مقدار area=12چاپ می شود.

ثابت : به مقداری که در طول اجرای یک برنامه تغییری نخواهد کرد ثابت یا constگفته می شود. بعد از مقدار دهی به یک ثابت شما نمی توانید

مقدار آن را تغییر دهید. ثابت ها در زبان جاوااسکریپت با کلمه کلیدی const تعریف می شوند.

```
const pi=3.14;
const month=12;
const hour=24;
```

اگر بعد از قطعه کد بالا دوباره به یکی از ثوابت مقدار جدیدی اختصاص بدهیم (برای مثال pi=4;) با خطای (Uncaught TypeError: Assignment to constant variable.) روبه رو می شویم.

فرق const, let و var چیست؟

وقتی متغیر ها را با var و let تعریف میکنیم می توانیم بعدا به آن ها مقدار بدهیم ولی وقتی یک متغیر با کلمه کلیدی const تعریف می شود همیشه باید همراه با تعریف متغیر مقدار ان را هم مشخص کنیم.

متغیر های let می توانند مقدار جدیدی بگیرند ولی نمی توانند دوباره تعریف شوند.

متغیر های var با هیچ مقدار یا undefined تعریف می شوند ، در صورتی که در let و const متغیر ها با هیچ مقدار پیاده سازی نمی شوند.

متغیر های var می توانند دوباره با var تعریف و همچنین مقدار دهی شوند.

متغیر های var یا دارای حوزه سراسری هستند یا حوزه تابعی . متغیر های let و const دارای حوزه بلاکی هستند.

تبدیل انواع داده به یکدیگر

اگر تبدیل نوع داده توسط خود زبان صورت بگیرد به آن تبدیل ضمنی گفته می شود و اگر توسط برنامه نویس انجام شود به آن تبدیل صریح گفته می شود .

در بیشتر مواقع عملگر ها و توابع به صورت خودکار مقادیر را به صورت صحیح به نوع دیگری تبدیل می کنند به این عمل تبدیل ضمنی نوع گفته می شود. برای مثال تابع alert به صورت خودکار هر داده ای که دریافت می کند را به نوع رشته تبدیل می کند و نمایش می دهد. متد () log هم هر مقداری که دریافت میکند را به رشته تبدیل میک ند و نمایش می دهد.

متد () prompt هم هر مقداری که دریافت میک ند را به رشته تبدیل میکند.

تبدیل های صریح

تبدیل به رشته

تابع ()String هر مقداری که به عنوان آرگومان دریافت می کند را به نوع رشته تبدیل می کند.

تبدیل به اعداد

تابع ()Number هر مقداری که به عنوان آرگومان دریافت می کند را به نوع رشته تبدیل می کند.

این تابع مقدار null را به صفر و مقدار undefined را به NaN تبدیل می کند.

تبدیل به Boolean

می توان این عملیات را به صورت صریح با استفاده از تابع Boolean() انجام داد.

مقادیری که معمولا به هیچ چیز اشاره دارند مثل null , undefined ، صفر و خالی به false و بقیه مقادیر به true تبدیل می شوند.

تبدیل به عدد اعشاری

با استفاده از تابع parseFloat() می توان آرگومان ورودی را به نوع داده عدد اعشاری تبدیل کرد

تبدیل به عدد صحیح

با استفاده از تابع parseInt() می توان آرگومان ورودی را به نوع داده عدد صحیح تبدیل کرد .

```
1.    <script>
2.    var num=2;
3.    let num2=31.14
4.    let Bool=true;
5.    let str="raman";
6.    console.log(parseFloat(num));
7.    console.log(parseInt(num2));
```

```
8.    console.log(parseInt(Bool));
9.    console.log(parseFloat(Bool));
10.   console.log(String(Bool));
11.   let str2=String(num2);
12.   console.log(str2);
13.   console.log(Number(str2));
14.   console.log(Number(str));
15.   console.log(Boolean(str));

16.   </script>
```

در قطعه کد بالا ابتدا چهار متغیر با انواع داده ی متفاوت تعریف شده اند. با استفاده از دستور خط 6 مقدار متغیر num به نوع اعشاری تبدیل می شود و نمایش داده می شود. با دستور خط 7 مقدار متغیر num2 که یک متغیر از نوع اعشاری است به نوع صحیح تبدیل می شود.

با دستور خط 8 مقدار متغیر از نوع منطقی به نوع عددی صحیح تبدیل می شود و نمایش داده می شود.

با دستور خط 9 مقدار متغیر Bool از نوع منطقی به نوع عددی اعشاری تبدیل می شود و نمایش داده می شود.

با دستور خط 10 مقدار متغیر Bool از نوع منطقی به نوع رشته تبدیل می شود و نمایش داده می شود.

دستور خط 11 یک متغیر جدید تعریف میکند به نام str2 که مقدار متغیر num2 از نوع عدد اعشاری تبدیل به رشته می شود و در آن ذخیره

می شود. دستور 12 این متغیر را چاپ می کند. دستور 13 این متغیر را به نوع عدد تبدیل میکند و نمایش می دهد.

دستور 14 و 15 به ترتیب متغیر str از نوع رشته را به نوع عددی و منطقی تبدیل می کنند و نمایش می دهند.

خروجی

2

31

NaN

NaN

true

31.14

31.14

NaN

True

```
1.    let str3="34samira";
2.    console.log(Number(str3));
```

در قطعه کد بالا یک متغیر از نوع رشته تعریف شده است و در دستور دوم به نوع عددی تبدیل شده و نمایش داده شده است. خروجی این کد NaN می باشد.

```
console.log(parseInt(str3));
```

خروجی دستور بالا عدد صحیح 34 می باشد. تابع parseInt از ابتدای رشته شروع به جستجو میک ند اگر اولین کاراکتر های آن عدد باشد آن ها را به عدد تبدیل می کند و با رسیدن به اولین کاراکتر از نوع رشته عمل تبدیل را متوقف میکند حتی اگر آخرین کاراکتر عدد باشد اگر در ابتدای رشته و قبل از کاراکتر ها نباشد به عدد تبدیل نمی شود.

منتظر کاربر بودن

متد prompt() یک جعبه مکالمه را نمایش می دهد که از کاربر درخواست یک مقدار ووردی را می کند. با استفاده از این جعبه قبل از ورودی کاربر به صفحه باید یک مقداری را وارد کند. تا هنگامی که این حعبه بسته نشود شما نمی توانید به دیگر قسمت های صفحه صفحه دسترسی داشته باشید.

هنگامی که این جعبه ظاهر می شود کاربر باید بر روی یکی از دکمه های ok cancel کلیک کند. در صورتی که کاربر بر روی دکمه okکلیک کند مقدار ورودی دریافت می شود و پنجره بسته می شود ، در غیر این صورت اگر کاربر دکمه cancel را انتخاب کند مقدار null برگشته می شود. مقدار برگشتی این تابع همیشه رشته است. اگر کاربر بدون وارد کردن چیزی دکمه ok را انتخاب کند یک رشته خالی دریافت می شود.

نحوه استفاده

```
prompt("text","defult text");
```

مقادیر پارامتر ها

پارامتر	type	توضیحات

text	string	**ضروری ، متنی است که در جعبه ی گفتگو نمایش داده می شود**
Defult text	string	**اختیاری ، متن ورودی پیش فرض را مشخص می کند**

مثال

```
1.   let name=prompt("what is your name?"
  ,"enter your name");
2.   let age=parseInt(prompt("how old are
   you?"));
3.   console.log("input name :",name);
4.   console.log("input age :",age);
```

در کد بالا ابتدا مقدار نام از نوع رشته از ورودی دریافت می شود و در متغیر name ذخیره می شود ، پیام اول یعنی "what is your name?" اجباری است ، زیرا کاربر با توجه به این پیام ورودی را وارد میکند اما پیام دوم اجباری نیست.

در دستور دوم مقدار سن با استفاده از متد prompt() به صورت رشته دریافت می شود ، سپس با استفاده از تابع parseInt() به نوع داده عددی صحیح تبدیل می شود و در متغیر age ذخیره می شود.

دستور سوم و چهارم به ترتیب name و age را چاپ می کنند.

مقادیر و انواع داده

در علم کامپیوتر، یک مقدار نمایشی از موجودیتی است که می تواند توسط یک برنامه دستکاری شود. اعضای یک نوع، مقادیر آن نوع هستند.یک مقدار یکی از موجودیت های بنیادی مانند یک حرف یا یک عدد است که برنامه با آن کار می کند. برنامه بدون مقدار نمی تواند کار کند. در زبان های که نوع داده آن ها پویاست متغیر ها محدود به نوع داده نیستند. یعنی یک متغیر می تواند در دستور اول نوع داده رشته ای داشته باشد و در دستور پنجم مقدار عددی دریافت کند و این کار بدون هیچ خطایی در زبان جاوااسکریپت ممکن است

مشاهده نوع یک مقدار در جاوااسکریپت

انواع داده در جاوا اسکریپت
زبان جاوااسکریپت 6 نوع داده دارد
نوع عددی (number)
زبان جاوااسکریپت فقط از یک نوع داده عددی پشتیبانی می کند و آن نوع داده Number و این به این معنی است که مانند دیگر زبان ها نوع داده Long و Int ندارد. هر عددی در جاوااسکریپت اعم از اعشاری و صحیح نوع داده Number است.

این نوع داده شامل هر دو نوع اعداد یعنی اعداد صحیح و اعشاری می باشد. در زبان جاوااسکریپت می توانیم بر روی اعداد هر عملیاتی را انجام دهیم : تقسیم بر صفر ، جمع ، ضرب و ... و هر گاه یک عملیات انجام نشدنی و بی معنی به مفسر بدهیم به مقدار NaN را برمی گرداند.
NaN متعلق به نوع عددی است و به معنای یک خطای محاسباتی است و نتیجه یک عملیات ریاضی اشتباه است. هر عملیاتی که روی مقدار NaN صورت بگیرد ننتیجه NaNخواهد داشت.
Infinityبیانگر مقدار بی نهایت است ، بی نهایت مقداری است بزرگ تر از هر عددی ، با تقسیم یک عدد بر صفر می توان به مقدار بی نهایت رسید. می توان مقدار بی نهایت را به صورت صریح در برنامه خود به کار برد.
در مثال زیر نتیجه هر عبارت رو به روی به صورت کامنت وجود دارد.

```
console.log(2+2); //4

console.log("samira"+12); //samira12

console.log("samira"*6); //Nan

console.log(3/0); //Infinity

console.log(23^2);//21

console.log(5%0);//NaN

console.log(12**2);//144
```

```
console.log(NaN+2);//NaN

console.log(Infinity);//Infinity
```

نوع داده رشته (String)

یک رشته مجموعه ای از چند حرف یا کاراکتر است که بین گیومه تک یا جفت قرار دارند .درزبان جاوااسکریپت فقط یک نوع داده رشته ای وجود دارد و آن هم string می باشد.

در زبان جاوا اسکریپت می توان رشته را به سه حالت داخل کوتیشن یا گیومه ها قرار داد:

دابل کوتیشن:”samira”

سینگل کوتیشن:’samira’

بک کوتیشن:`samira`

مورد اول و دوم رشته های ساده ای هستند اما مورد دوم به ما امکان می دهد که متغیر ها را برای ارزیابی و محاسبه درون رشته ها قرار دهیم.

در مثال زیر نتیجه هر عبارت رو به روی به صورت کامنت وجود دارد.

```
let num=3;
    console.log("samira",num); //samira 3

    console.log('number is ${num}');//numb
er is ${num}

    console.log('number is $num'); //numbe
r is $num

    console.log(`number is ${num}`);//numb
er is 3

    console.log(`result ${3} + ${5} = ${5}
  `);//result 3 + 5 = 5
```

نوع داده آرایه

دستورات و عبارات

عملگر ها و عملوند ها

عملگر های محاسباتی

عملگر های مقایسه ای

عملگر های انتساب

عملگر های بیتی

عملگر های منطقی

اولویت عملگر ها

ترکیب

متد های کنسول در جاوااسکریپت