**متغیرها**

بیشتر اوقات برنامه‌های جاوا اسکریپت نیاز دارند تا با اطلاعات کار کنند. بطور نمونه :

1 یک فروشگاه آنلاین - اطلاعات شامل کالاهایی که فروخته می‌شوند و سبد خرید بشود.

2 یک نرم‌افزار چت - اطلاعات شامل کاربران، پیام‌ها و دیگر موارد می‌شود.

متغیرها برای ذخیره سازی اطلاعات استفاده می‌شوند.

**متغیر**

یک متغیر در واقع فضایی برای ذخیره سازی اطلاعات با نامی مشخص است. ما از متغیرها برای ذخیره سازی نام کالاها، بازدید‌کنندگان و دیگر انواع اطلاعات استفاده می‌کنیم.

برای ساخت یک متغیر در جاوا اسکریپت از let استفاده کنید.

عبارت زیر یک متغیر با نام message تعریف میکند. به این عمل define یا declare گفته می‌شود.

کد

حالا می‌توانیم اطلاعاتی را از طریق عملگر = در این متغیر قرار دهیم.

کد

حال رشته کاراکترهایی که نوشتیم در بخشی از حافظه کامپیوتر که در اختیار این متغیر می‌باشد ذخیره شده است.

کد

به منظور احتصار می‌توانیم تعریف متغیر و مقداردهی به آن را در یک خط قرار دهیم.

کد

همینطور می‌توانیم چند متغیر را در یک خط تعریف کنیم.

کد

این مورد ممکن است کوتاه‌تر به نظر آید اما ما پیشنهاد نمی‌کنیم. چراکه خوانایی را پایین می‌آورد. لطفا برای تعریف متغیرها از خطوط جداگانه استفاده کنید.

کد

بعضی برنامه‌نویسان متغیرهای متعدد را به این صورت در چند خط می‌نویسند :

کد

همینطور برخی به صورت comma-first اینکار را انجام می‌دهند :

کد

از لحاظ فنی همه این حالات کار خواهند کرد و انتخاب آن‌ها به سلایق و زیبایی شناسی شخصی مربوط است.

**Var بجای let**

در اسکریپت‌های قدیمی ممکن است بجای کلیدواژه let با کلیدواژه var برخورد کنید.

کد

کلیدواژه var تقریبا با let یکسان است و یک متغیر را تعریف می‌کند، ولی روش کار آن قدیمی است.

تفاوت کوچکی بین این دو وجود دارد که در حال حاضر به آن نمی‌پردازیم و در مبحث The old "var"به آن خواهیم پرداخت.

**یک مقایسه‌ی عینی \*\*\***

برای درک مفهوم متغیر، می‌توانیم آن را یک جعبه برای نگهداری اطلاعات تصور کنیم، که یک نام منحصربفرد روی آن چسبانده‌ایم.

برای نمونه، تصور کنید متغیر message در جعبه‌ای با برچسب message و با مقدار Hello! باشد :

تصویر

ما می‌توانیم هر مقداری در آن قرار دهیم. همینطور می‌توانیم هر چند بار که خواستیم این مقدار با تغییر دهیم.

کد

وقتی مقدار تغییر کرد، اطلاعات قبلی از آن پاک خواهند شد :

تصویر

ما همینطور می‌توانیم دو متغیر تعریف کرده و اطلاعات یکی را در دیگری کپی کنیم.

کد

**زبان‌های Functional**

جالب است بدانید زبان‌های Functional مانند Scala یا Erlang تغییر مقدار متغیر را ممنوع کرده‌اند.

در این نوع زبان‌ها وقتی مقداری را در جعبه قرار می‌دهیم تا ابد آنجا می‌ماند و اگر بخواهیم مقداری دیگر را ذخیره کنیم، ما را مجبور می‌کنند تا جعبه‌ای جدید بسازیم.

اگرچه در نگاه اول این زبان‌ها عجیب به نظر می‌رسند، ولی کاملا توانایی توسعه‌ (در پروژه‌های) جدی را دارند. مهم‌تر از آن در زمینه‌هایی مانند محاسبات موازی (Parallel Computation) این محدودیت‌ها تبدیل به مزیت می‌شوند. مطالعه‌ی چنین زبان‌هایی (حتی اگر به این زودی قصد استفاده از آنها را ندارید) برای وسیع شدن دیدتان، توصیه می‌شود.

**نام‌گذاری متغیرها**

دو محدودیت برای نام‌گذاری متغیرها در جاوا اسکریپت وجود دارد :

1 نام باید فقط از کاراکترها، اعداد یا علامت $ و \_ تشکیل شده باشد.

2 اولین کاراکتر نمی‌تواند عدد باشد.

بطور نمونه :

کد

زمانی‌که به یک نام متشکل از چند کلمه است، عموما از روش camelCase استفاده می‌شود. به این صورت که کلمات یکی پس از دیگری پشت هم نوشته می‌شوند و حرف اول هر کلمه کاراکتر بزرگ هست : myVeryLongName .

همینطور از $ و \_ نیز می‌توان استفاده نمود. معنی خاصی ندارند و همانند حروف قابل استفاده هستند.

این نام‌ها معتبر هستند :

کد

این نام‌ها نامعتبر هستند :

کد

**بزرگی و کوچکی حروف مهم است**

متغیر apple با Apple متفاوت است.

**کاراکترهای غیر انگلیسی مجاز هستند ولی توصیه نمی‌شوند**

امکان استفاده از کاراکترهای زبان‌های دیگر وجود دارد. مانند :

کد

از نظر فنی اینها درست کار می‌کنند ولی بر اساس یک رسم بین‌المللی برای نام متغیرها از زبان انگلیسی استفاده می‌شود. حتی اگر اسکریپتی کوچک می‌نویسیم، ممکن است تا مدت طولانی‌ای مورد استفاده و توسعه قرار بگیرد و اشخاصی از سایر کشورها ممکن است نیاز باشد روزی آن اسکریپت را بخوانند.

**نام‌های رِزِرو شده**

تعدادی نام رزرو شده وجود دارد که نمی‌توان از آنها برای نام متغیرها استفاده نمود چراکه آنها توسط خودِ زان جاوا اسکریپت استفاده شده‌اند.

برای نمونه let، class، return و function رزرو هستند.

این کد ارور Syntax خواهد داشت :

کد

**مقداردهی بدون use strict**

در گذشته این امکان وجود داشت تا متغیری را بدون استفاده از let و صرفا با مقداردهی تعریف نمود. درصورتیکه از use strict استفاده نکنیم، این امکان همچنان وجود دارد،

کد

این روش مناسب نیست و در صورت وجود use strict با خطا مواجه خواهد شد.

کد

**ثوابت**

برای تعریف ثابت (متغیری که تغیر نمی‌کند) از const بجای let استفاده می‌کنیم :

کد

متغیرهایی که با const ساخته می‌شود را ثابت (constant) می‌گوییم. آنها قابل تغییر نیستند و تلاش برای اینکار با خطا روبرو خواهد شد :

کد

وقتی برنامه‎‌نویس اطمینان دارد که متغیری هیچگاه تغییر نمی‌کندُ می‌تواند آن را به عنوان const تعریف کند تا برای دیگران نیز این موضوع واضح باشد.