**انواع داده**

یک متغیر در جاوا اسکریپت می‌تواند هر نوع داده‌ای را در خود ذخیره کند. یک متغیر می‌تواند در یک لحظه حاوی رشته‌ای اِز کاراکترها و در لحظه دیگر یک عدد باشد.

کد

زبان‌های برنامه‌نویسی‌ای که چنین امکانی به شما می‌دهند Dynamically Typed نامیده می‌شوند. به این معنی که انواعی از داده وجود دارد و متغیرها محدود به آنها نیستند.

در جاوا اسکریپت هفت نوع پایه‌ای از انواع داده وجود دارد که الان به صورت کلی به آنها می‌پردازیم و در بخش‌های بعدی در مورد هرکدام با جزئیات صحبت خواهیم کرد.

**عدد**

کد

نوع number (عدد) اعداد صحیح و اعداد اعشاری را شامل می‌شود. عملیات مختلفی در مورد اعداد وجود دارد مانند ضرب (\*) ، تقسیم (/) ، جمع (+) ، تفریق (-) و غیره.

همینطور بجز اعداد معمولی، اعداد خاصی نیز وجود دارند که به همین نوع از متغیرها مربوط می‌شوند. یعنی : بینهایت، - بینهایت و NaN .

بینهایت : نماد بیانگر علامت ریاضیاتیِ ∞ است. این مقدار خاص بوده که از هر عدد دیگری بزرگتر است.

از تقسیم هر عددی با صفر به این مقدار می‌رسیم :

کد

یا به طور مستقیم نیز به آن دسترسی داریم :

کد

NaN: بیانگر یک اشکال محاسباتی است و در نتیجه یک عملیات ریاضیاتی غلط یا تعریف نشده بوجود می‌آید. برای نمونه:

کد

هر عملی بر روی NaN نتیجه NaN خواهد داشت :

کد

در نتیجه اگر NaN در عملیات ریاضیاتی‌ای وجود داشته باشد، بر روی تمام معادله تاثیر می‌گذارد (نتیجه معادله برابر NaN خواهد بود).

**عملگرهای ریاضیاتی امن هستند**

عملیات ریاضی در جاوا اسکریپت امن است. ما هر نوع عملی می‌توانیم انجام دهیم مانند تقسیم بر صفر. اسکریپت ما هیچگاه با خطا مواجه نخواهد شد. در بدترین حالت NaN را به عنوان نتیجه خواهیم گرفت.

**رشته کاراکترها**

یک رشته کاراکتر در جاوا اسکریپت باید در بین کوتِیشِن‌ها محصور شوند.

کد

در جاوا اسکریپت سه نوع کوتِیشِن داریم :

Double qoutes مانند "Hello".

Single qoutes مانند 'Hello'.

Backticks مانند `Hello`.

Double quotes و Single quotes همان کوتِیشِن‌های عادی هستند و در جاوا اسکریپت تفاوتی با هم ندارند.

Backticks امکان توسعه به ما می‌دهند. بوسیله‌ی آنها می‌توانیم داخل یک رشته کاراکتر عبارات و دستورات جاوا اسکریپت بنویسیم.

کد

عبارتی که در ${…} قرار می‌گیرد اجرا شده و نتیجه آن در رشته مورد نظر قرار می‌گیرد.

توجه داشته باشید که quote ها چنین قابلیتی را ندارند :

*کد*

در مورد رشته‌ها در بخش‌های بعدی بیشتر صحبت خواهیم کرد.

**نوع داده‌ای برای کاراکترها وجود ندارد**

در زبان‌هایی مانند C و یا Java نوع داده‌ای خاصی مختص به کاراکترها تحت عنوان char وجود دارد.

در جاوا اسکریپت چنین نوعی نداریم. فقط یک نوع داده برای رشته‌ها داریم که آن string است، که شامل یک یا چند کاراکتر می‌تواند باشد.

**نوع Boolean**

نوع Boolean فقط یکی از دو مقدار true و false را شامل می‌شود.

این نوع معمولا برای ذخیره مقدار yes یا no استفاده می‌شود. True به معنی yes و false به معنی no می‎‌باشد.

برای نمونه :

کد

مقدار Booealn معمولا به عنوان یک نتیجه‌ی مقایسه بدست می‌آیند :

کد

در مورد Boolean ها در بخش‌های بعدی صحبت خواهیم کرد.

**مقدار null**

مقدار null یکی دیگر از انواع داده در جاوا اسکریپت می‌باشد.

کد

در جاوا اسکریپت null مانند بعضی از زبان‌های برنامه‌نویسی به معنی وجود نداشتن یک شیء یا به معنی null pointer نیست.

Null صرفا یک مقدار خاص است که نمایانگر "خالی بودن" ، "هیچ بودن" و "مشخص نبودن مقدار" می‌باشد.

در مثال بالا مقدار age به هر دلیلی نا مشخص یا خالی است.

**مقدار undefined**

مقدار undefined نیز مانند null یک نوع مجزا در جاوا اسکریپت است.

Undefined بدین معنی‌ است که "مقداری اختصاص نیافته است".

اگر متغیری تعریف کنیم و مقداری به آن اختصاص ندهیم، مقدار آن undefined خواهد بود :

کد

از لحاظ فنی، امکان تخصیص دادن undefined به هر متغیری وجود دارد :

کد

اما ما چنین کاری را توصیه نمی‌کنیم و برای تخصیص مقدار "خالی" یا "مجهول" از null استفاده می‌کنیم و از undefined برای بررسی اینکه به متغیری مقدار تخصیص یافته یا خیر استفاده می‌کنیم.

**Object ها و Symbol ها**

نوع object از انواع خاص است.

انواعی که تا بدین جا مطالعه کردیم از انواع "اولیه" بودند چراکه مقدار آنها فقط شامل یک چیز می‌شد. اما object ها برای ذخیره مجموعه‌ای از داده‌ها به شکلی پیچیده‌تر استفاده می‌شوند. پس از آنکه در مورد انواع اولیه بیشتر مطالعه کردیم، در مورد object ها بیشتر خواهیم آموخت.

نوع symbol برای ایجاد یک شناسه منحصر به فرد برای object ها مورد استفاده قرار می‌گیرد. به منظور تکمیل عنوان این بخش این نوع داده را نیز در اینجا عنوان کردیم ولی بهتر است جزئیات آن را در بخش‌های بعدی بررسی کنیم.

**عملگر typeof**

این عملگر نوع عبارت را نشان می‌دهد. معمولا زمانی که می‌خواهیم سریع نوع یک مقدار را بررسی کنیم یا زمانی‌که می‌خواهیم پردازشی بر اساس نوع یک مقدار انجام دهیم، کاربرد دارد.

این عملگر به دو شکل قابل استفاده است :

به عنوان عملگر : typeof x

به عنوان فانکشن : typeof (x)

با صدا زدن typeof x رشته کاراکتری حاوی نوع آن مقدار نمایش داد می‌شود :

کد

سه خط آخر احتمالا نیاز به توضیحات بیشتری دارد.

1 Math یک آبجکت در اصطلاحا built-in (از پیش نوشته شده در هسته زبان) است که عملیات متنوع ریاضیاتی در اختیار ما قرار می‌دهد. در اینجا صرفا یک مثال از آن را نمایش دادیم و در بخش Numbers با آن بیشتر آشنا خواهیم شد.

2 خروجی typeof null همانطور که می‌بینید object است و این صحیح نیست. این یک خطا در نوع کار typeof می‌باشد که به منظور سازگاری باقی مانده است. مطمئنا null یک object نیست. خودِ null یکی از انواع داده در جاوا اسکریپت است.

3 نوع alert یک فانکشن است چراکه alert یکی از فانکشن‌های جاوا اسکریپت می‌باشد. ما در بخش‌های بعدی با فانکشن‌ها بیشتر آشنا خواهیم شد و خواهیم آموخت که نوعی تحت عنوان فانکشن نداریم و خودِ فانکشن‌ها در اصل از نوعِ object هستند. اما عملگر typeof به آنها به طرز دیگری برخورد می‌کند.

**خلاصه**

در جاوا اسکریپت 7 نوع داده‌ی پایه‌ای داریم :

number که شامل اعداد صحیح و اعداد اعشاری می‌شود.

string برای رشته کاراکترها. یک string می‌تواند یک یا چند کاراکتر داشته باشد. هیچ نوع داده‌ای تحت عنوان char در جاوا اسکریپت وجود ندارد.

boolean برای true و یا false.

null برای مقادیر مجهول. null یک نوع مستقل بوده که یک مقدار تحت عنوان null دارد.

undefined برای مقادیر تخصیص نیافته. یک نوع مستقل بوده که یک مقدار تحت عنوان undefined دارد.

object برای ذخیره ساختارهای پیچیده‌تر اطلاعات.

symbol برای شناسه‌های یکتا.

عملگر typeof به شما اجازه می‌دهد نوع مقدار ذخیره شده در یک متغیر را تشخصی دهید.

به دو طریق : typeof x و یا typeof(x)

نوع متغیر را باز می‌گرداند مانند "string"

برای null مقدار "object" را نمایش می‌دهد. این یک خطا در خودِ زبان است. در واقع null یک object نیست.