

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

انواع داده data types در جاوا اسکریپت

مدرس: مهندس افشین رفوآ

انواع داده در جاوا اسکریپت

نوع داده دسته بندی ای از انواع تایپ‌های موجود مانند اعداد صحیح (*integer*)، بولی (*boolean*) و... است که ویژگی‌هایی از جمله : مقادیر قابل پذیرش توسط متغیر، عملیات قابل اجرا بر روی متغیر، معنای داده و حافظه مورد نیاز برای ذخیره متغیر را تعیین می کند.

انواع داده : رشته، عدد، بولی، آرایه، شی (*object*)

نوع داده در جاوا اسکریپت

جاوا اسکریپت می تواند *data type* های گوناگونی مانند : اعداد (نوع عددی)، رشته (نوع رشته ی)، آرایه (از نوع آرایه) و از نوع *object* (شی).

```
var length = 16;           // Number
var lastName = "Johnson"; // String
var cars = ["Saab", "Volvo", "BMW"]; // Array
var x = {firstName:"John", lastName:"Doe"}; // Object
```

مفهوم نوع داده

در برنامه نویسی، نوع داده مفهوم بسیار مهمی است.

برای اینکه بتوان روی متغیرها عملیات انجام داد، باید چیزی درباره ی نوع آن دانست.

بدون نوع داده، کامپیوتر قادر به فهم و محاسبه صحیح و امن داده نیست.

```
var x = 16 + "Volvo";
```

آیا عاقلانه به نظر می رسد که "Volvo" را به عدد 16 اضافه کنیم؟ آیا نتیجه ی صحیح و منطقی بدست می دهد و یا در نتیجه ی عملیات خطا برمی گرداند (ارور می دهد)؟

جاوا اسکریپت مثال بالا را اینگونه پردازش می کند.

```
var x = "16" + "Volvo";
```

در صورتی که عملوند / operand دوم یک رشته باشد، جاوا اسکریپت عملوند اول را نیز رشته محسوب می کند.

مثال

```
var x = 16 + "Volvo";
```

جاوا اسکریپت عبارات را از چپ به راست ارزیابی می کند. ترتیب و توالی متفاوت ممکن است نتایج متفاوتی را رقم بزند.

```
var x = 16 + 4 + "Volvo";
```

نتیجه

20Volvo

مثال جاوا اسکریپت

```
var x = "Volvo" + 16 + 4;
```

نتیجه

Volvo164

در مثال اول، جاوا اسکریپت 16 و 4 را به عنوان رقم یا عدد در نظر می گیرد تا اینکه به "Volvo" می رسد. (که یک رشته تلقی می گردد)

حال در مثال دوم، از آنجایی که عملوند اول یک رشته است، دیگر عملوندها نیز به مثابه ی رشته در نظر گرفته می شوند.

جاوا اسکریپت دارای نوع داده ی پویا (dynamic type) است.

جاوا اسکریپت نیز دارای نوع داده ی پویا است، بدین معنا که یک متغیر را می توان به عنوان نوع های مختلف بکار برد.

مثال

```
var x; // Now x is undefined
var x = 5; // Now x is a Number
var x = "John"; // Now x is a String
```

رشته ها در جاوا اسکریپت (متغیر رشته ای)

یک رشته / *string* (یا همان رشته ی مبتنی بر متن / *text string*) متشکل است از یک سری کاراکتر مانند این مثال "John Doe" که یک رشته محسوب می شود.

رشته ها باید حتماً داخل علامت نقل و قول نوشته شوند. برای این منظور می توان از هر دو کاراکتر " و ' بهره گرفت.

مثال

```
var carName = "Volvo XC60"; // Using double quotes
var carName = 'Volvo XC60'; // Using single quotes
```

همچنین می توان از علامت نقل و قول داخل رشته استفاده کرد، البته تا زمانی که با علامت نقل و قول اطراف رشته یکی نشوند.

مثال

```
var answer = "It's alright"; // Single quote inside double quotes
var answer = "He is called 'Johnny'"; // Single quotes inside double quotes
var answer = 'He is called "Johnny"'; // Double quotes inside single quotes
```

اعداد در جاوا اسکریپت (متغیرهای عددی)

جاوا اسکریپت تنها یک نوع عدد دارد، به عبارتی دیگر اعداد را می توان با اعشار (*decimal*) یا بدون اعشار بکار برد / متغیرهای عددی در جاوا اسکریپت یا به صورت اعشاری و دهدی نوشته می شوند و یا به صورت عدد صحیح نمایش داده می شوند.

مثال

```
var x1 = 34.00; // Written with decimals
var x2 = 34; // Written without decimals
```

اعدادی که بیش از حد بزرگ یا کوچک هستند را می توان با نماد علمی / **scientific notation** نوشت.

```
var y = 123e5;      // 12300000
var z = 123e-5;    // 0.00123
```

Boolean / نوع داده ی بولی (متغیر بولی) در جاوا اسکریپت

متغیرهای بولی تنها می توانند دو مقدار **true** یا **false** را داشته باشند.

مثال

```
var x = true;
var y = false;
```

متغیرهای بولی برای آزمون و تست یک شرط بکار می رود.

آرایه ها (arrays) در جاوا اسکریپت

آرایه ها در جاوا اسکریپت داخل کروشه یا همان کاراکتر **[]** قرار می گیرند.

آیتم های یک آرایه توسط ویرگول از هم جدا می شوند.

کد زیر آرایه ی به نام **cars** که شامل سه آیتم (اسم اتومبیل) می باشد را ایجاد (تعریف) می کند.

مثال

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

اندیس یا شاخص آرایه از صفر آغاز می گردد، بدین معنا که آرایه ی اول **[0]**، دومین **[1]** و به همین ترتیب ادامه پیدا می کند.

اشیا (objects) در جاوا اسکریپت

در جاوا اسکریپت اشیا داخل کاراکتر **{}** (آکولاد) قرار داده و تعریف می شوند. خصوصیت های اشیا در جاوا اسکریپت داخل آکولاد به صورت جفت اسم : مقدار (**name:value pairs**) نوشته شده و توسط ویرگول از هم جدا می گردند.

مثال

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

شی (person) در مثال فوق دارای چهار خصوصیت می باشد: **firstName**، **lastName**، **age** و **eyeColor**.

عملگر typeof

می توان با استفاده از عملگر **typeof** نوع یک متغیر جاوا اسکریپت را بدست آورد / عملگر ذکر شده نوع متغیر را برمی گرداند.

مثال

```
typeof "John"           // Returns string
typeof 3.14             // Returns number
typeof false           // Returns boolean
typeof [1,2,3,4]       // Returns object
typeof {name:'John', age:34} // Returns object
```

نکته

در جاوا اسکریپت، آرایه در واقع یک نوع شی محسوب می گردد. بنابراین **typeof [1,2,3,4]** در نتیجه نوع **object** را برمی گرداند.

Undefined (تعریف نشده)

متغیری که مقداردهی نشده است، به صورت پیش فرض تعریف نشده تلقی می گردد و مقدار آن **undefined** می باشد. در صورت استفاده از **typeof** نیز مقدار **undefined** به عنوان نتیجه بازگردانده می شود.

مثال

```
var person;           // Value is undefined, type is undefined
```

هر متغیری را می توان با تنظیم مقدار آن روی **undefined** تهی کرد، همچنین نوع آن **undefined** یا تعریف نشده در نظر گرفته می شود.

مثال:

```
person = undefined; // Value is undefined, type is undefined
```

مقادیری تهی / empty values

یک مقدار تهی هیچ گونه ارتباطی با مقدار تعریف نشده (**undefined**) ندارد.

یک متغیر رشته‌ی تهی (**empty string variable**) هم مقدار و هم نوع آن تعریف شده هستند. (دارای نوع و مقدار است)

مثال

```
var car = ""; // The value is "", the typeof is string
```

null

Null در جاوا اسکریپت به معنای تهی یا "**nothing**" است. متأسفانه در زبان برنامه نویسی بیان شده نوع داده **null** یک شی حساب می‌شود.

با تنظیم مقدار یک شی روی **null** می‌توان آن شی را تهی کرد / تخصیص **null** به شی می‌توان آن را تهی کرد.

مثال

```
var person = null; // Value is null, but type is still an object
```

همچنین می‌توان برای تهی کردن یک شی آن را **undefined** مقداردهی کرد.

مثال

```
var person = undefined; // Value is undefined, type is undefined
```

در مثال زیر تفاوت بین **undefined** و **null** تشریح شده

```
typeof undefined // undefined
typeof null       // object
null === undefined // false
```

```
null == undefined // true
```

www.tahlildadeh.com