بسم اللّه الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

عبارات با قاعده / JavaScript – Regular expressions

مدرس: مهندس افشین رفوآ

آدرس آموزشگاه. تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 561 - واحد 7
88146323 - 88446780 - 88146330

عبارات با قاعده / JavaScript – Regular expressions

Regular expression (عبارت منظم یا باقاعده) رشته یا توالی از کاراکترها است که یک یگ الگو جستجو (search pattern) تعریف می کند. برای انجام عملیات جستجو برای کلمات و یا کاراکترهای مورد نظر در متن یک صفحه می توان از این عبارات کمک گرفت .

الگوی جستجو را می توان برای انجام عملیاتی همچون جستجو در متن و جایگزینی نوشته هایی در متن مورد استفاده قرار داد.

عبارت با قاعده (regular expression) چیست؟

Regular expression، شی ای است که درباره ی **الگوی کاراکترها** شرح می دهد.

زمانی که در یک متن به دنبال داده یا اطلاعات معینی به جستجو می پردازید، می توانید از یک الگوی خاص برای کلمه ی مورد جستجو استفاده نمایید.

یک کاراکتر تنها، می تواند یک الگوی ساده باشد.

یک الگوی پیچیده تر ممکن است شامل کاراکترهای بیشتری باشد و از آن می توان برای اجرای عملیاتی همچون تجزیه کردن، قالب دهی، بررسی کردن و جایگزینی استفاده کرد.

دستور نگارش (syntax)

```
/pattern/modifiers;
```

مثال

```
var patt = / tahlildadeh /i;
```

توضیح مثال

/w3schools/i یک عبارت با قاعده است.

w3schools نیز یک یک الگو است که در جستجو به کار می رود.

i یک modifier می باشد (باعث می شود جستجو به کوچک بزرگی حروف حساسیت نشان ندهد).

استفاده از توابع رشته ای (string methods)

عبارات با قاعده در جاوا اسکریپت اغلب با دو متد رشته ای search() و replace() بکار می رود.

Search () – جهت جستجو برای کشف وجود یا عدم وجود یک کلمه خاص در یک متغیر رشته ای بکار گرفته می شود.

Replace () – از این متد به منظور جستجو به دنبال یک حرف یا کلمه خاص در یک متغیر رشته ای و جایگزینی آن با یک مقدار جدید استفاده می شود.

مثال

با استفاده از یک RegEx یک جستجو که نسبت به کوچک و بزرگی حروف حساسیت ندارد به دنبال "w3schools" در رشته انجام دهید.

```
var str = "Visit W3Schools";
        var n = str.search(/w3schools/i);
```

نتیجه

6

تابع جستجو (search method) همچنین می تواند یک رشته به عنوان آرگومان جستجو بپذیرد. آرگومان گفته شده به یک عبارت با قاعده (regular expression) تبدیل می شود.

مثال

با استفاده از یک رشته به عنوان آرگومان ورودی تابع search () برای یافتن واژه ی "W3schools" در string جستجو انجام می دهیم.

```
var str = "Visit W3Schools!";
        var n = str.search("W3Schools");
```

در این مثال با استفاده از تابع replace () یک عبارت با قاعده ی case-insensitive (W3Schools) را جایگزین کلمه ی Microsoft می کنیم.

```
var str = "Visit Microsoft!";
        var res = str.replace(/microsoft/i, "tahlildadeh");
```

نتیجه

```
Visit W3Schools!
```

تابع replace() نیز می تواند مانند متد search () یک رشته به عنوان آرگومان ورودی می پذیرد.

```
var str = "Visit Microsoft!";
        var res = str.replace("Microsoft", " tahlildadeh");
```

اگر به مثال های بالا خوب دقت کرده باشید متوجه می شوید که

می توان به جای string argument (آرگومان رشته ای) از آرگومان های regular expression در توابع یاد شده استفاده کرد.

با بهره گیری از عبارات با قاعده می توان جستجو را قوی تر یا به مراتب کارآمدتر ساخت(برای مثال آن را نسبت به کوچک بزرگی حروف بی تفاوت کرد).

تعریف / تنظیم کننده های نحوه ی جستجو (modifier) عبارات با قاعده

Modifier یک کاراکتر است که با استفاده از آن جستجو به دنبال کلمه یا حرف مد نظر را case-insensitive (بی تفاوت به کوچک و بزرگی حروف) کرده یا آن را سراسری (global) می کنیم.

Modifier	شرح
i	بیانگر case-insensitive (حساس نبودن به کوچک و بزرگی حروف) بودن جستجو است.
g	نشانگر سراسری بودن جستجو می باشد (جستجو پس از یافتن مورد اول متوقف نمی شود).
m	جستجو را در چندین خط انجام می دهد.

الگوی عبارات با قاعده (regular expression pattern)

Pattern عبارتی است که در متن می خواهیم به دنبال آن بگردیم.

با استفاده از کاراکتر [] می توان مجموعه ای از کاراکترها را جستجو کرد (یافت)

عبارت	شرح
[abc]	هر کاراکتری که درون (محصور در) [] باشد را پیدا می کند.
[0-9]	هر عددی که بین براکت باز و بسته محصور شده باشد را پیدا می کند.
(x|y)	هر کاراکتر ی(جایگزین) که با " | " از هم جدا شده باشند را پیدا می کند.

Metacharacter کاراکتری است که معنای به خصوصی برای برنامه ی کامپیوتری داشته باشد.

Metacharacter	شرح
\d	یک عدد را پیدا می کند.
\s	یک کاراکتر خط فاصله را می یابد.
\b	مورد مد نظر (match) را در ابتدا یا انتهای یک کلمه می یابد.
\uxxxx	کاراکتر Unicode که توسط عدد مبنای 16 (hexadecimal) xxxx مشخص شده را می یابد.

Quantifiers (کمیت سنج ها) مقادیر یا کمیت را تعیین می کنند.

Quantifier	توضیح
n+	با رشته ای match می شود که دربردانده ی حداقل یک n باشد.
n*	با رشته ای match می شود که دربردارنده ی 0 یا بیشتر تکرار (نمونه) از n باشد.

با رشته ای match می شود که دربردارنده ی 0 یا بیشتر تکرار (نمونه) از n باشد.	**n?**

استفاده از شی RegExp

شی ای است دارای توابع و خواص از پیش تعریف شده ای است.

متد test()

این تابع در یک رشته به دنبال الگوی موردنظر جستجو انجام داده سپس بسته به نتیجه، مقدار true یا false باز می گرداند. به عبارتی دیگر، متد test()، یک رشته را داخل یک مقدار مشخص جستجو کرده و براساس نتیجه، true یا false برمی گرداند.

مثال زیر در یک رشته به دنبال "e" می گردد.

مثال

```
var patt = /e/;
        patt.test("The best things in life are free!");
```

به این خاطر که "e" در رشته ی مورد نظر یافت می شود، خروجی کد true خواهد بود.

نیازی نیست عبارت با قاعده را درون یک متغیر قرار دهید. دو خط کد مثال بالا را می توان در یک خط (به طور خلاصه) نوشت.

```
/e/.test("The best things in life are free!");
```

تابع exec()

این متد داخل یک رشته به دنبال الگوی مورد نظر جستجو انجام داده، سپس نوشته ی یافت شده را بر می گرداند یا به عبارت دیگر متد exec()، یک رشته را داخل یک مقدار مشخص جستجو می کند و با توجه به نتیجه ی حاصل، مقدار جستجو شده یا false را بازیابی می کند.

حال در صورتی که مورد (match) یافت نشد، متد ذکر شده null باز می گرداند.

مثال زیر در رشته به دنبال کاراکتر "e" می گردد.

```
/e/.exec("The best things in life are free!");
```

به این خاطر که "e" در رشته ی مورد نظر یافت می شود، خروجی کد e خواهد بود.