

## ساختار کد

@alithecodeguy

اولین مبحثی که راجع به آن صحبت خواهیم کرد ، ساختار یک قطعه کد است .

### جملات (statements)

statement ها ساختار گرامری و دستوراتی هستند که عمل خاصی را انجام می دهند . تا کنون statement زیر را دیده ایم :

```
alert('Hello World');
```

که از آن برای نمایش پیغام Hello World استفاده می شود . هر تعداد statement که بخواهیم می توانیم در کد خود داشته باشیم . statement ها می توانند بوسیله ؛ از یکدیگر جدا شوند . برای مثال :

```
alert('Hello'); alert('World');
```

برای خوانایی بیشتر کد ، معمولا statement ها را در خطوط جداگانه ای می نویسند .

```
alert('Hello');
```

```
alert('World');
```

### سمی کالن ها (semicolons)

برای جداسازی خطوط ، در اکثر اوقات می توانیم semicolon ها را ننویسیم . به عنوان مثال قطعه کد زیر به درستی کار می کند :

```
alert('Hello')
```

```
alert('World')
```

در این مثال ، جاوا اسکریپت خطوط را توسط semicolon های ضمنی از یکدیگر جدا می کند .  
به این عمل automatic semicolon insertion می گویند .

در اکثر اوقات ، یک خط جدید به یک semicolon اشاره دارد ، ولی اکثر اوقات ، نه همیشه !

در مثال زیر ، خط جدید به معنای semicolon نیست :

```
alert(3 +
```

```
1
```

```
+ 2);
```

@alithecodeguy

خروجی قطعه کد بالا ۶ می باشد چرا که جاوا اسکریپت در انتهای خطوط از semicolon استفاده نکرده است . به صورت بصری مشهود است که هرگاه خط با علامت + به اتمام رسید ، آن عبارت ناتمام بوده و به semicolon نیاز نمی باشد .

مواردی هم وجود دارد با اینکه به semicolon نیاز است ولی جاوا اسکریپت آنگونه که مورد انتظار است ، عمل نمی کند .

پیدا کردن و رفع اینگونه خطاها معمولا بسیار سخت است .

به مثال زیر توجه کنید :

```
[1, 2].forEach(alert)
```

در حال حاضر نیازی نیست که نگران معنی [] و forEach باشید . در آینده راجع به آنها صحبت خواهیم کرد . در حال حاضر فقط در نظر داشته باشید که خروجی عبارت فوق نمایش عدد ۱ و عدد ۲ می باشد . حال قبل از عبارت فوق یک alert دیگر اضافه می کنیم و در انتهای آن semicolon نمی گذاریم :

```
alert("There will be an error")
```

```
[1, 2].forEach(alert)
```

حال اگر قطعه کد بالا را اجرا کنید ، فقط **alert** ابتدایی نمایش داده می شود و سپس به خطا می خوریم ولی هنگامی که در انتهای خط اول **semicolon** بگذاریم همه چیز درست می شود.

```
alert("There will be an error");
```

```
[1, 2].forEach(alert)
```

@alithecodeguy

حال اگر کد را اجرا کنیم ابتدا متن اولی نمایش داده می شود سپس اعداد ۱ و ۲ نمایش داده می شود.

دلیل خطای فوق این بود که جاوا اسکریپت قبل از علامت براکت **[]** نمی تواند به طور ضمنی **semicolon** وارد کند.

بنابراین قطعه کد خطا دار را به این شکل می بیند :

```
alert("There will be an error")[1, 2].forEach(alert)
```

در صورتی که باید دو **statement** مجزا باشند . ترکیب هایی مانند این مثال ، منجر به خطا می شود . چنین خطاهای در سایر موارد نیز ممکن است اتفاق بیفتد .

پیشنهاد می شود حتی در صورتی که **statement** ها در خطوط مجزایی قرار دارند ، باز هم در انتهای هر خط **semicolon** را قرار دهید . این قانون توسط اکثر برنامه نویسان رعایت می گردد . یک بار دیگر تاکید می کنیم : در اکثر اوقات می توانیم از گذاشتن **semicolon** پرهیز کنیم ولی پیشنهاد می شود که همیشه از آن استفاده کنید مخصوصا اگر تازه کار هستید .

## کامنت ها (comments)

با گذشت زمان ، برنامه ها پیچیده و پیچیده تر می شوند و اضافه کردن **comment** هایی که چرایی و چگونگی کد را شرح دهند ضروری تر می گردد .

**comment** ها می توانند در هر قسمت از اسکریپت گذاشته شوند . آنها تاثیری در اجرای برنامه ندارند چرا که **engine** ها آنها را نادیده می گیرند .

**comment** های یک خطی با علامت **//** آغاز می شوند و باقی خط به عنوان **comment** در نظر گرفته می شود که ممکن است یک خط کامل را اشغال کند یا در انتهای یک **statement** قرار گیرد . مثال :

```
// This comment occupies a line of its own
```

```
alert('Hello');
```

```
alert('World'); // This comment follows the statement
```

**comment** های چند خطی با علامت **/\*** / آغاز شده و با علامت **\*/** تمام می شوند .

مثال :

```
/* An example with two messages.
```

```
This is a multiline comment.
```

```
*/
```

```
alert('Hello');
```

```
alert('World');
```

@alithecodeguy

محتویات بین این دو علامت نادیده گرفته می شود و تاثیری در اجرا ندارد .

همچنین می توانیم برای غیر فعال کردن موقت قسمتی از کد هم از آن استفاده کنیم.

استفاده کردن از hotkeys

در اکثر ویرایشگر ها ، می توان با استفاده از دکمه های ترکیبی `Ctrl + /` یک خط و با استفاده از دکمه های ترکیبی `Ctrl+Shift+/` چند خط را کامنت کرد.  
در Mac به جای دکمه `Ctrl` از `Cmd` و به جای `Shift` از `Option` استفاده کنید.

کامنت ها تو در تو

نمی توانید از کامنت های تو در تو استفاده نمایید و کدی مانند کد زیر منجر به خطا می شود :

```
/*
```

```
/* nested comment !?! */
```

```
*/
```

```
alert( 'World' );
```

لطفا در استفاده از `comment` خساست به خرج ندهید. `comment` ها به طور کلی تعداد خط کد را بالا می برند ولی اصلا مشکلی نیست. ابزار های زیادی وجود دارد که می توانند قبل از منتشر کردن کد ، این خطوط را حذف نمایند.  
در بخش های بعد در مورد `code quality` صحبت خواهیم کرد که بخشی از آن به موضوع نوشتن کامنت های بهتر اختصاص می یابد.

@alithedeguy