بسم اللّه الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

روش ها و رهنمود های کد نویسی در جاوا اسکریپت

مدرس : مهندس افشین رفوآ

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

روش ها و رهنمود های کد نویسی در جاوا اسکریپت

روش ها یا رهنمود های کدنویسی، دستور هایی برای کدنویسی صحیح و بهینه هستند که شامل موارد زیر می باشند.

1. قواعد مربوط به تعریف متغیر و توابع
2. قوانین مربوط به استفاده از خط فاصله، تو رفتگی (indet) و توضیحات (comment)
3. اصول و روش های برنامه نویسی

رهنمود های کدنویسی کیفیت و کارایی بهینه ی کد را تضمین می کنند.

1. خوانایی کد را بهبود می بخشد.
2. نگهداشت (maintenance) کد را آسان می سازد.

رهنمودهای کد نویسی می توانند روش های کد نویسی شخصی و مختص به خود شما باشند یا قوانین مستند و ثبت شده باشند که یک تیم برنامه نویسی از آن پیروی می کند.

قوانین مربوط به تعریف اسم برای متغیرها و توابع

توصیه می کنیم برای تعریف اسم متغیرها و توابع از camelCase استفاده کنید.

همچنین توصیه می کنیم تمامی اسم ها با یک حرف شروع شوند.

نمونه

```
firstName = "John";
        lastName = "Doe";
        price = 19.90;
        tax = 0.20;
        fullPrice = price + (price * tax);
```

استفاده از فضای خالی در اطراف عملگرها

توصیه می نماییم همیشه قبل و بعد از عملگرها (= + - * /) و نیز پس از ویرگول یک فضای خالی ایجاد کنید.

نمونه

```
var x = y + z;
        var values = ["Volvo", "Saab", "Fiat"];
```

تنظیم تورفتگی کد

همیشه به اندازه ی چهار space، کد خود را تو بگذارید (indent کنید)

```
function toCelsius(fahrenheit) {
            return (5 / 9) * (fahrenheit - 32);
        }
```

توجه: توصیه می کنیم از کلید tab برای تو گذاری استفاده نکنید، زیرا که برنامه های ویرایش گر مختلف ممکن است آن را به گونه ای متفاوت تفسیر کرده و اجرا کنند.

قواعد مربوط به نوشتن و استفاده از دستورات ساده و مرکب

1. همیشه دستورات ساده ی خود را با کارکتر نقطه ویرگول " ; " خاتمه دهید.

```
  var values = ["Volvo", "Saab", "Fiat"];
          var person = {
              firstName: "John",
              lastName: "Doe",
              age: 50,
              eyeColor: "blue"
          };
```

قوانین کلی برای دستورات پیچیده (مرکب)

2. همیشه براکت باز " { " در انتهای اولین خط برنامه ی خود قرار دهید.
3. قبل از براکت باز یک فضای خالی ایجاد کنید.
4. براکت بسته را در یک خط جدید و بدون هیچ فضای خالی قبل از آن قرار دهید.
5. به هیچ وجه یک دستور مرکب را با کاراکتر نقطه ویرگول خاتمه ندهید.

مثال

```
function toCelsius(fahrenheit) {
            return (5 / 9) * (fahrenheit - 32);
        }
        for (i = 0; i < 5; i++) {
            x += i;
        }
        if (time < 20) {
            greeting = "Good day";
        } else {
            greeting = "Good evening";
        }
```

قوانین مربوط به تعریف شی

1. براکت باز را در همان خطی که اسم object در آن قرار دارد، تایپ کنید.
2. بین هر خاصیت (property) و مقدار آن یک دو نقطه گذاشته و قبل و بعد آن فاصله ایجاد کنید.
3. تنها مقادیر رشته ای را داخل علامت (" ") محصور کنید، از بکار بردن نقطه ویرگول در اطراف مقادیر عددی خودداری کنید.
4. پس از آخرین جفت خاصیت – مقدار (property-value) در تعریف یک شی از ویرگول استفاده نکنید.
5. براکت بسته را در خط جدیدی قرار داده و پیش از آن فضای خالی قرار ندهید.
6. همیشه تعریف یک object را با کاراکتر نقطه ویرگول خاتمه دهید.

مثال

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

می توان آن دسته از object هایی ک تعریف آن ها کوتاه است را در یک خط واحد گنجاند، بدین شکل

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

طول خط

به منظور خوانایی بهتر کد، توصیه می کنیم تعداد کاراکترهای خط کد خود را به 80 کاراکتر محدود کنید.

چنانچه یک دستور جاوا اسکریپت به قدری طولانی است که در یک خط جا نمی شود، در آن صورت توصیه می کنیم آن خط را به دو بخش تقسیم کنید. بهترین مکان برای شکستن خط کد پس از یک عملگر یا یک کاراکتر ویرگول می باشد.

مثال

```
document.getElementById("demo").innerHTML =
            "Hello Dolly.";
```

قوانین تعریف اسم برای توابع و متغیرها

در تعریف یا انتخاب اسم برای متغیرها و توابع ثابت قدم باشید، بدین معنا که

1. همیشه از سیستم نگارشی camelCase برای تعریف اسم متغیرها و توابع استفاده کنید.
2. اسم متغیرهای سراسری را با حروف بزرگ " UPPERCASE " نگارش کنید.
3. ثابت ها (همچون PI) را با حروف بزرگ " UPPERCASE " تایپ کنید.

آیا می توان از hyp-hen، camelCase یا under_score در اسم متغیرها استفاده کرد؟

استفاده از Hyphen در HTML و CSS

خصیصه های HTML5 می توانند با data- آغاز شوند (data-quantity، data-price).

CSS از hyphen در جفت های خاصیت–مقدار (property-name) استفاده می کند، مانند: (font-size).

توجه: استفاده از hyphen در جاوا اسکریپت مجاز نمی باشد، به این خاطر که ممکن است با عملگر تفریق اشتباه گرفته شود.

Underscore

اغلب برنامه نویسان استفاده از " _ " را به ویژه در زبان SQL ترجیح می دهند (date_of_birth).

از Underscore بیشتر در زبان PHP استفاده می شود.

PascalCase

این سیستم نگارشی بیشتر توسط برنامه نویسان C بکار گرفته می شود.

CamelCase

نگارش camelCase نیز توسط خود جاوا اسکریپت و کتابخانه های آن از جمله jQuery مورد استفاده قرار می گیرد.

نکته

هیچگاه اسامی توابع و متغیرها را با علامت $ آغاز نکنید. این کار باعث می شود با اسم هایی که در کتابخانه های جاوا اسکریپت بکار می روند، تداخل ایجاد شود.

بارگذاری (اسکپریت های خارجی) جاوا اسکریپت در صفحه ی HTML

به منظور بارگذاری اسکریپت های خارجی کافی است از یک ساختار نگارشی ساده استفاده کنید، برای مثال نیازی به استفاده از خصیصه ی type در اسکریپت نیست.

```
<script src="myscript.js">
```

دسترسی به المان های HTML

کدنویسی نامرتب در HTML منجر به بروز خطاهایی در نتایج تولید شده از اسکریپت های اجرا شده ی جاوا اسکریپت می گردد. به عنوان مثال می توان به دو دستور جاوا اسکریپت زیر توجه کرد که هر دو نتیجه ای متفاوت بدست می دهند.

```
var obj = getElementById("Demo")
        var obj = getElementById("demo")
```

در صورت امکان سعی کنید از همان روش های نام گذاری توابع و متغیرها که برای جاوا اسکریپت شرح داده شد، در HTML استفاده کنید.

پسوند فایل ها

فایل های HTML باید دارای پسوند .html باشند (استفاده از .htm غیرمجاز می باشد).

فایل های CSS باید پسوند .css را داشته باشند.

فایل های دربردارنده ی اسکریپت های جاوا اسکریپت باید دارای پسوند .js باشند.

استفاده از حروف کوچک برای نام گذاری فایل ها

بیشتر web server ها از جمله Apache و Unix نسبت به کوچک و بزرگی اسم فایل حساس هستند.

london.jpg با London.jpg یکسان نیستند و برای دسترسی به london.jpg نمی توان از London.jpg استفاده کرد.

در حالی که برخی از web server ها همچون Microsoft و IIS نسبت به کوچک و بزرگی حروف حساس نیستند.

به عنوان مثال جهت دسترسی به London.jpg می توان از london.jpg نیز استفاده کرد.

چنانچه از ترکیبی از حروف کوچک و بزرگ استفاده می کنید باید ثابت قدم باشید.

اگر web server خود را از یکی که نسبت به کوچک و بزرگی حروف حساس نیست به یک سرور که به کوچک و بزرگی حروف حساس است تغییر دهید، حتی کوچکترین اشتباه هم منجر به از کار افتادگی و خرابی وب سایت شما می شود.

برای اجتناب از بروز خطاهایی از این دست، توصیه می کنیم همیشه از حروف کوچک استفاده کنید.

تاثیر در کارایی

قواعد و روش های ذکر شده در اجرای برنامه شما تاثیر چندانی ندارد.

تورفتگی کدها و فضاهای خالی در اسکریپت های کوتاه تقریبا نادیده گرفته می شوند.

برای کدی که در حال شکل گرفتن است، قابلیت خوانایی از بیشترین اهمیت برخوردار است. توصیه می کنیم اسکریپت ها بزرگتر را کوچک سازی کنید.