

# JavaScript Core

## JavaScript Core

### دوره دستورات جاوا اسکریپت

نحوه آدرس دهی فایل جاوا اسکریپت

```
<script src="./js/main.js"></script>
```

نمایش مقادیر

```
document.write("hi");  
alert('hi');  
console.log('hi');
```

تعریف متغیر

```
var personName="mohammad";  
console.log(personName);
```

شرط

```
var name = 'mohammad';  
var age = 55;  
  
if(age > 30){  
    document.write('ok');  
}else document.write('nok');
```

تابع نویسی

```
function sum(num1 , num2){  
    return num1+num2;  
}  
document.write(sum(1,2));
```

تمرین اول: دو ورودی از کاربر بگیرد و بگوید که کدام بزرگتر است

in main.js:

```
function compertion(){
    input = Number(document.getElementById("input").value);
    output = Number(document.getElementById("output").value);
    if (input > output){
        return console.log("input>output");
    }else if (input == output){
        return console.log("input=output");
    }else return console.log("input<output");
}
```

in index.html:

```
<body>
    <p>Input</p>
    <input type="text" id="input"/>
    <p>output</p>
    <input type="text" id="output"/><br>
    <p><button onclick="compertion()">Run</button></p>
    <script src="./js/main.js"></script>
</body>
```

آرایه ها

```
var profileInfo = ['ali', 'mohammad', 'javad', 'taghi'];
profileInfo[3]="0"
profileInfo.push(true);
profileInfo.unshift("username");
profileInfo.pop();
profileInfo.shift();
console.log(profileInfo.indexOf('javad'));
console.log(profileInfo);
```

ساخت یک object

```
var profileInfo = {
    name : "ali",
    lastname : "mohammadi",
    skills : ['developer', 'driving', 'studing'],
    age : function(yearOfBirth){
        return 2019 - yearOfBirth;
    }
};
profileInfo["name"] = 'javad';
```

```
console.log(profileInfo.skills[0]);
console.log(profileInfo.age(1994));
```

حلقه ها

```
var skills = ['developer', 'driving', 'studing'];

for (i=0 ; i < skills.length ; i++){
    console.log(skills[i]);
}
```

```
var i=0;
var skills = ['developer', 'driving', 'studing'];

while (i< skills.length){
    console.log(skills[i]);
    i++;
}
```

## DOM (Document Object Model)

دسترسی به المان های صفحه HTML

```
var show = document.getElementById("app");

var show2 = document.getElementsByClassName("box");

var show3 = document.getElementsByTagName("p");
```

خروجی سه دستور بالا به صورت آرایه می باشد

برنامه ای که محتوی خط اول را عوض نماید . برای خط دوم لینک تنظیم کند. برای خط سوم لینک را عوض کرده و برای خط چهارم رنگ را عوض کند.  
کد صفحه آن به صورت زیر است :

```
<div id="main">
    <p>Change content</p>
    <a href="#" target="_blank">setlinkaddress</a>
    <a href="http://mohammadkhalatbari.ir" target="_blank">Change
link address</a>
    <p>Change color of paragraph</p>
    <button onclick="change()">click to change</button>
</div>
```

```
function change(){
    var main = document.getElementById("main");
    main.getElementsByTagName("p")[0].innerHTML = "new content";
    main.getElementsByTagName("a")[0].href = "http://google.com";
    main.getElementsByTagName("a")[1].setAttribute("href" ,
"http:reactjs.com");
    main.getElementsByTagName("p")[1].style.color = 'red';
}
```

ساخت تگ ساخت ساخت کانتنت تخصیص کانتنت به تگ بعد هم تخصیص این تگ داخل دیو

```
<div id="show"></div>
```

javascript

```
//create tag
var pTag = document.createElement("p");

//content tag
var text = document.createTextNode("this is a paragraph");

//append content to tag
pTag.appendChild(text);

var divTag = document.getElementById("show");
divTag.appendChild(pTag);

//divTag.removeChild(pTag);
```

بدون استفاده از onclick رنگ یک متن را تغییر دهد و ساعت را نمایش دهد

```
<p id="p1">change color</p>
<p id="p2">click to display date</p>
```

javascript

```
var p1 = document.getElementById("p1");

// add event Listener
p1.addEventListener("click" , changeColor);
```

```
function changeColor(){
    p1.style.color = "red"
}

var p2 = document.getElementById("p2");
p2.addEventListener("click" , displayDate);

function displayDate(){
    console.log(Date());
}
```

دریافت و انجام کار بر روی تمام تگ های یکسان یک صفحه

```
var pList = document.querySelectorAll("p");
```

## ES6

---

ساخت متغیر و متغیر های ثابت

```
let x = 5;
const age = 29;
```

تغییر روش تعریف توابع (arrow function)

```
var sumFunction = (a,b) => {
    var sum = Number(a) + Number(b);
    return sum;
}

console.log(sumFunction(5,6));
```

قابلیت default parameter تنها در ES6 امکان پذیر می باشد.

```
var sumFunction = (a,b=7) => {
    var sum = Number(a) + Number(b);
    return sum;
}

console.log(sumFunction(5));
```

در ساخت object میتوان مقدار value به key ندهیم و در بیرون object مقدار دهی کنیم.

```
let name = "Ali";
let obj = {
  name,
  age : 25
};
console.log(obj.name);
```

میتوان مقدار key را هم string داد فقط فراخوانی آن متفاوت است

```
let obj = {
  "name" : "Javad",
  age : 25
};
console.log(obj["name"]);
```

فقط در ES6 میتوان مقدار key را خارج object تعریف کرد. فقط به مقدار دهی دقت شود.

```
let email = "email";
let obj = {
  "name" : "Javad",
  age : 25,
  [email] : "a@gmail.com"
};
console.log(obj["name"]);
```

تعریف حلقه for برای object ها

```
let i ;
let array = ["ali" , 29 , "developer"]
for (i in array){
  console.log(array[i]);
}
```

استفاده از متغیرها درون تعریف string فقط دقت شود به جای "" از `` استفاده میکنیم.

```
let name = "Ali";
let jobTitle = ` ${name}
is Developer `;
console.log(jobTitle);
```

تعریف آرایه ها نیز کمی تغییر کرده است

```
let profile = ["ali" , 29 , "developer"];
let [a,b] = profile;
console.log(a,b);
```

برای object ها نیز همانند آرایه ها می باشد با کمی تفاوت.

```
let profile = {
  name : "ali" ,
  age : 29,
  job : "developer"
};
let {name , age } = profile;
console.log(name , age);
```

در صفحات وب بسیار مهم است که فایل های زیاد را import نکنیم که سرعت لود صفحه کاهش یابد. برای این منظور بهتر است که import را بصورت جزئی تر انجام دهیم. همانند فایل زیر :

```
var sumFunction = (a,b=7) => {
  var sum = Number(a) + Number(b);
  return sum;
}
export { sumFunction };
//////////new file
import {sumFunction} from './main.js'
```

## OOP in JavaScript

```
class profileInfo{
  constructor(name, age, email){
    this.name = name;
    this.age = age;
    this.email = email;
  }

  displayEmail(){
    console.log(this.email);
  }
}

let Mohammad = new profileInfo("mohammad" , 29 , "a@gmail.com");
Mohammad.displayEmail();
```

خط آخر نحوه ساخت Object از روی کلاس می باشد.

نحوه ارث بری

```
class ProfileInfo{
    constructor(name, age, email){
        this.name = name;
        this.age = age;
        this.email = email;
    }

    displayEmail(){
        console.log(this.email);
    }
}

class Developer extends ProfileInfo{
    constructor (name,age,email,skills){
        super(name,age,email);
        this.skills=skills;
    }
}

let Mohammad = new Developer("mohammad" , 29 , "a@gmail.com" ,
"javascript");
Mohammad.displayEmail();
```