جلسه سوم web3

1. **معرفی جاوا اسکریپت و محل نوشتن آن**
2. **معرفی تگ** **Script**
3. **معرفی متغیر ها و مفهوم local , global**
4. **معرفی انواع داده**
5. **معرفی فانکشن در فضای HTML**
6. **معرفی alert و console**
7. **معرفی onClick**
8. **معرفی add event listener**
9. **متغیرها**: معرفی var, let, و const و تفاوت‌های آنها.

 **Global Variables**: در تمام بخش‌های برنامه قابل دسترسی هستند.

 **Local Variables**: فقط در محدوده تعریف شده (مثل توابع یا بلوک‌ها) قابل دسترسی هستند.

**let** و **const**: در محدوده بلوک تعریف شده‌اند و با ویژگی‌های مختلفی از جمله قابلیت تغییر (let) و غیر قابل تغییر (const).

 **var**: در محدوده تابع تعریف شده و به بالای محدوده‌اش منتقل می‌شود (هوئیستینگ).

1. Scope (محدوده دسترسی)
2. Hoisting (بالا کشیدن)
3. بازتعریف و تغییر مقدار
4. **انواع داده‌ها**: معرفی انواع داده‌های اصلی مثل اعداد، رشته‌ها،

تفاوت اصلی آرایه و آبجکت :

آرایه‌ها :

آرایه‌ها در جاوااسکریپت نوع داده‌ای قدرتمند و انعطاف‌پذیر هستند که به شما امکان می‌دهند مجموعه‌ای از داده‌ها را به صورت ساختاریافته نگهداری و مدیریت کنید. با استفاده از متدها و توابع مختلف، می‌توانید عملیات‌های پیچیده‌ای بر روی آرایه‌ها انجام دهید و داده‌های خود را به شکل موثری مدیریت کنید.

اشیاء.:

* **شیء (Object)**: یک نوع داده ساختاریافته است که جفت‌های کلید-مقدار را ذخیره می‌کند.
* **کلید**: یک رشته (String) است که به یک مقدار اشاره می‌کند.
* **مقدار**: می‌تواند هر نوع داده‌ای باشد.
* **متد**: یک تابع است که به عنوان مقدار در شیء تعریف شده است.

اشیاء در جاوااسکریپت برای نگهداری و مدیریت داده‌ها بسیار کاربردی هستند و بخش اساسی از برنامه‌نویسی شیءگرا را تشکیل می‌دهند.

### تفاوت‌های کلیدی

* **اندیس‌گذاری vs کلیدها**: در آرایه‌ها از اندیس‌های عددی برای دسترسی به عناصر استفاده می‌شود، در حالی که در اشیاء از کلیدهای رشته‌ای (یا Symbol) استفاده می‌شود.
* **مناسب برای نوع داده‌ها**: آرایه‌ها برای لیست‌های مرتب و توالی داده‌ها مناسب هستند، در حالی که اشیاء برای نگهداری داده‌های مرتبط با یک موجودیت (مانند اطلاعات شخصی) مناسب‌تر هستند.
* **عملیات و متدها**: آرایه‌ها دارای متدهای زیادی برای کار با لیست‌ها هستند، مانند push, pop, map, filter و غیره. اشیاء نیز می‌توانند متد داشته باشند، اما به طور کلی برای دسترسی و تغییر داده‌ها به صورت جفت‌های کلید-مقدار استفاده می‌شوند.

1. **مفهوم notation در جاوا اسکریپت : رعایت با این موارد:**

Notation به معنای استفاده از استانداردهای نگارش و سبک‌های کدنویسی است. این استانداردها شامل نام‌گذاری متغیرها، توابع، استفاده از فاصله‌ها، تورفتگی‌ها (indentation)، و سایر قواعدی است که به خوانایی و سازمان‌دهی کد کمک می‌کند.

* 1. نام‌گذاری متغیرها و توابع
  2. استفاده از فاصله‌ها و تورفتگی‌ها (Indentation)
  3. استفاده از نقطه‌ویرگول (Semicolons)
  4. فاصله‌ها (Spaces)
  5. استفاده از پرانتزها و آکولادها (Parentheses and Braces)
  6. کامنت‌گذاری (Comments)
  7. استفاده از Camel Case برای نام‌گذاری متغیرها و توابع

1. **چرا فایل‌های جاوا اسکریپت خارجی را استفاده کنیم؟**
2. **سازماندهی بهتر**: نگهداری کدهای جاوا اسکریپت در فایل‌های جداگانه به شما کمک می‌کند تا کدهای HTML و جاوا اسکریپت خود را جدا و مرتب نگه دارید.
3. **استفاده مجدد**: شما می‌توانید یک فایل جاوا اسکریپت را در چندین صفحه HTML مختلف استفاده کنید.
4. **افزایش سرعت بارگذاری**: مرورگرها می‌توانند فایل‌های جاوا اسکریپت را کش (cache) کنند، که این باعث افزایش سرعت بارگذاری صفحات می‌شود.

**روش استفاده از فایل‌های جاوا اسکریپت خارجی**

برای درج یک فایل جاوا اسکریپت خارجی در سند HTML، از تگ <script> با ویژگی src استفاده می‌کنیم.

1. **انواع عملگرها: انجام عملیات برروی مقادیر**

**1. عملگرهای حسابی (Arithmetic Operators)**

این عملگرها برای انجام عملیات ریاضی استفاده می‌شوند:

* **جمع و تفریق (+)**: دو مقدار را با هم جمع می‌کند.

let a = 5;

let b = 3;

let result = a + b; // نتیجه: 8

* **ضرب (\*)**: و تقسیم (/)دو مقدار را در هم ضرب می‌کند.

jlet result = a \* b; // نتیجه: 15

* **باقیمانده (%)**: باقیمانده تقسیم مقدار اول بر مقدار دوم را محاسبه می‌کند.

let result = a % b; // نتیجه: 2

**2. عملگرهای انتساب (Assignment Operators)**

این عملگرها برای انتساب مقدار به یک متغیر استفاده می‌شوند:

* **انتساب ساده (=)**: مقدار سمت راست را به متغیر سمت چپ انتساب می‌دهد.

javascript

Copy code

let x = 10; // مقدار 10 به متغیر x انتساب داده شده است

**3. عملگرهای مقایسه‌ای (Comparison Operators)**

این عملگرها برای مقایسه دو مقدار استفاده می‌شوند و نتیجه آنها یک مقدار بولی (Boolean) است:

* **برابر (==)**: بررسی می‌کند که آیا دو مقدار برابر هستند یا خیر.

let isEqual = (a == b); // نتیجه: false

* **برابر دقیق (===)**: بررسی می‌کند که آیا دو مقدار و نوع آنها برابر هستند یا خیر.

let isStrictEqual = (a === b); // نتیجه: false

* **نامساوی (!=)**: بررسی می‌کند که آیا دو مقدار برابر نیستند یا خیر.

let isNotEqual = (a != b); // نتیجه: true

* **بزرگتر (>)**: بررسی می‌کند که آیا مقدار اول بزرگتر از مقدار دوم است یا خیر.

let isGreater = (a > b); // نتیجه: true

* **کوچکتر (<)**: بررسی می‌کند که آیا مقدار اول کوچکتر از مقدار دوم است یا خیر.

let isLesser = (a < b); // نتیجه: false

**4. عملگرهای منطقی (Logical Operators)**

این عملگرها برای عملیات منطقی استفاده می‌شوند:

* **و (&&)**: بررسی می‌کند که آیا هر دو شرط درست هستند یا خیر.

let x = true;

let y = false;

let result = (x && y); // نتیجه: false

* **یا (||)**: بررسی می‌کند که آیا حداقل یکی از شرایط درست است یا خیر.

let result = (x || y); // نتیجه: true

* **نفی (!)**: معکوس مقدار بولی را برمی‌گرداند.

let result = !x; // نتیجه: false

**5. عملگرهای افزایشی و کاهشی (Increment and Decrement Operators)**

این عملگرها برای افزایش یا کاهش مقدار یک متغیر استفاده می‌شوند:

* **افزایشی (++)**: کاهشی (--) مقدار متغیر را یک واحد افزایش می‌دهد.

let count = 0;

count++; // اکنون count برابر با 1 است

### معرفی DOM (Document Object Model) در جاوا اسکریپت

**DOM** یا **مدل شیء سند**، یک واسط برنامه‌نویسی برای اسناد HTML و XML است. DOM ساختار یک سند را به صورت یک درخت از گره‌ها (Nodes) نمایش می‌دهد که هر گره نماینده یک قسمت از سند است، مانند عناصر (Tags)، ویژگی‌ها (Attributes) و متن.

#### مفهوم ساده

می‌توان DOM را مانند یک نقشه درختی از صفحه وب تصور کرد. با استفاده از این نقشه، جاوا اسکریپت می‌تواند به عناصر مختلف صفحه وب دسترسی پیدا کند، آنها را تغییر دهد، حذف کند و یا عناصر جدیدی اضافه کند.

#### اجزاء DOM

1. **عناصر (Elements)**: نماینده تگ‌های HTML هستند.
2. **ویژگی‌ها (Attributes)**: نماینده ویژگی‌های تگ‌های HTML هستند.
3. **متن (Text)**: محتوای متنی داخل تگ‌های HTML.

#### چرا DOM مهم است؟

* **دسترسی**: جاوا اسکریپت می‌تواند به تمام عناصر صفحه وب دسترسی پیدا کند.
* **تغییر**: می‌توان محتوا، استایل و ساختار صفحه را تغییر داد.
* **تعامل**: می‌توان به تعاملات کاربر پاسخ داد، مانند کلیک روی دکمه‌ها.

1. **پاسخ به سوالات دانشجویان.**

* تزریق در جلسات اول : مدیریت زمان و برنامه نویسی رفتار روزانه
* حل معما
  1. **هر جلسه مواردی که در جلسه قبل یاد گرفتن نوشته باشن و شرح بدن**

**تمرین عملی کلاس**

1. **معرفی innerHTMLو innerText**
2. **معرفی عملگرها**
3. **معرفی شرط if**