

Программирование на стороне клиента

Модуль 2 / 5

Технологии

The logo for JavaScript, featuring the letters "JS" in a bold, black, sans-serif font, centered within a solid yellow square.

JS



Подключение стилей в HTML

По аналогии со стилями, скрипты подключаются либо как внешний файл через атрибут `src` тега `script`, либо пишутся прямо в теге

```
<script src="js/script.js"></script>  
<script>  
|    //А можно прямо здесь  
</script>
```

Порядок загрузки

Скрипты, подключенные в секции `head`, будут загружены до основного контента страницы, а перед закрывающим тегом `body` – после основного контента. Но:

С атрибутом `defer` – будут выполняться после загрузки страницы, независимо от места подключения

С атрибутом `async` – могут выполняться до загрузки страницы, независимо от места подключения

Типы данных

В JavaScript есть 8 основных типов.

- `number` для любых чисел: целочисленных или чисел с плавающей точкой; целочисленные значения ограничены диапазоном $\pm(2^{53}-1)$.
- `bigint` для целых чисел произвольной длины.
- `string` для строк. Строка может содержать ноль или больше символов, нет отдельного символьного типа.
- `boolean` для `true/false`.
- `null` для неизвестных значений – отдельный тип, имеющий одно значение `null`.
- `undefined` для неприсвоенных значений – отдельный тип, имеющий одно значение `undefined`.
- `object` для более сложных структур данных.
- `symbol` для уникальных идентификаторов.

```
>> typeof 2
< "number"
>> typeof 'Hello'
< "string"
>> typeof true
< "boolean"
>> typeof {}
< "object"
>> typeof []
< "object"
>> typeof 4.4
< "number"
>> typeof NaN
< "number"
>> typeof null
< "object"
>> typeof undefined
< "undefined"
>> |
```

Объявление переменных

Переменные объявляются с помощью ключевых слов `let`, `var`, `const`. Как правило – `let`. Стоит обратить внимание, что `let` имеет блочную область видимости, а `const` не меняет значение

<https://medium.com/nuances-of-programming/var-let-const-%D0%B2-%D1%87%D0%B5%D0%BC-%D1%80%D0%B0%D0%B7%D0%BD%D0%B8%D1%86%D0%B0-dc6c3beb0b12>

Типизация динамическая, то есть тип данных определится в зависимости от того, что будет присвоено в качестве значения

```
<script>

  let intNumber = 2

  let floatNum = intNumber * 4.3

  let bool = true

  let str = 'Hello JS!!!'

</script>
```

Логирование в консоль

Функция `console.log()` выводит переданное значение в консоль. Полезно при отладке.

```
let a = 2 * 5  
  
console.log(a)
```

10

task1.html:20:17

>>



Математические операции

Математические операции выполняются обычно, как и во всех языках.

% - остаток от деления

Класс Math – для математических операций, таких как возведение в степень, извлечение корня, округление и т.д.

```
let a = -2, b = 3, c = 4
```

```
let discriminant = Math.pow(b, 2) - 4 * a * c
```


Строки

```
"Lorem" + "Ipsum"  
"LoremIpsum"  
`Now is ${new Date().toString()}`  
"Now is Sat Feb 12 2022 21:27:04 GMT+0600 (Восточный Казахстан)"  
|
```

```
>> "Hello, ToU!".substring(0, 5)  
← "Hello"  
  
>> "Hello, ToU!".split(" ")  
← ▶ Array [ "Hello,", "ToU!" ]  
  
>> "Hello, ToU!".replaceAll("Hello", "Good bye")  
← "Good bye, ToU!"  
  
>> "Hello, ToU!".toLowerCase()  
← "hello, tou!"  
  
>> "Hello, ToU!".toUpperCase()  
← "HELLO, TOU!"  
  
>> "Hello, ToU!".indexOf("ToU")  
← 7  
  
>> "Hello, ToU!".includes("Hello")  
← true  
  
>>
```

Объекты

Поля объекта могут быть простыми типами, а могут быть другими объектами или массивами:

```
let human = {  
  name: "Alina",  
  age: 21,  
  position: "UX designer",  
  education: [  
    {  
      institution: "Business College",  
      graduationYear: 2019  
    },  
    {  
      institution: "Toraighyrov University",  
      graduationYear: 2022  
    }  
  ]  
}  
  
console.log(human.name) // Alina
```

Массивы

```
let languages = ["Python", "Java", "JavaScript", "C#", "PHP"]  
  
languages.push("Go")  
  
languages.forEach(item => console.log(item))
```

Python	task1.html:22:43
Java	task1.html:22:43
JavaScript	task1.html:22:43
C#	task1.html:22:43
PHP	task1.html:22:43
Go	task1.html:22:43

Обращение к элементу массива идет по индексу, например:

languages[2] – выведет JavaScript

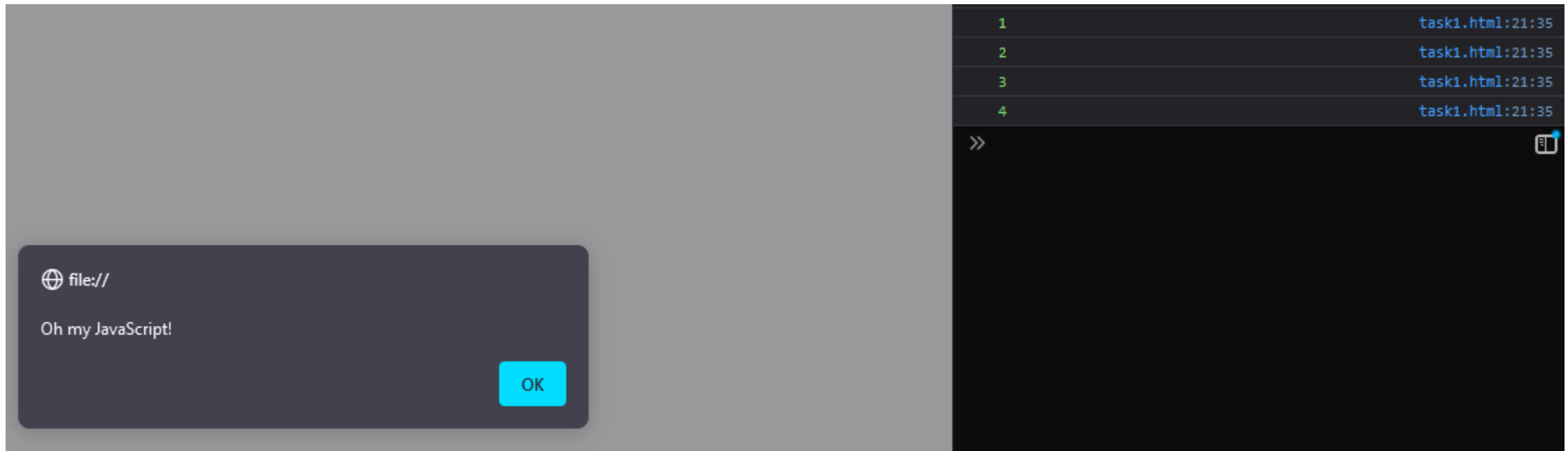
Нумерация массива начинается с нуля

Все методы работы с массивами:

<https://learn.javascript.ru/array-methods>

Таймаут и интервал

```
setTimeout(() => alert("Oh my JavaScript!"), 5000)  
  
let i = 1  
setInterval(() => console.log(i++), 1000)
```



The screenshot shows a web browser interface. In the foreground, an alert dialog box is displayed with the text "Oh my JavaScript!" and an "OK" button. In the background, the browser's developer console is open, showing a list of log messages. The messages are numbered 1 through 4, each with a timestamp of "task1.html:21:35". The first message is "1", the second is "2", the third is "3", and the fourth is "4". The console also shows a ">>" prompt and a small icon in the bottom right corner.

Message	Timestamp
1	task1.html:21:35
2	task1.html:21:35
3	task1.html:21:35
4	task1.html:21:35

Манипуляции с DOM

В данном примере скрипт каждую секунду обновляет содержимое элемента.

```
<div class="container py-5">
|   <h1 id="counter" class="text-center"></div>
</div>

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

  const counterElement = document.getElementById('counter')

  let i = 1
  setInterval(() => counterElement.innerHTML = i++, 1000)

</script>
```

Получение элемента

Функции, возвращающие коллекцию элементов, можно перебрать через `.forEach`, как и массивы.

Функции `.querySelector` и `.querySelectorAll` принимают такие же селекторы, как и `css`

```
const element = document.getElementById('identity') //вернет 1 элемент
const byClass = document.getElementsByClassName('item') //вернет 1 элемент
const byClassAll = document.getElementsByClassName('item') //вернет список элементов
const bySelector = document.querySelector('.list .item') //вернет 1 элемент
const bySelectorAll = document.querySelectorAll('.list .item') //вернет список элементов

element.children // Потомки
element.parentElement // Родитель
element.nextElementSibling // Последующий элемент
element.previousElementSibling // Предыдущий элемент

let div = document.createElement('div') // Создает новый

element.appendChild(div) // Добавляет дочерний элемент
```

Манипуляции с содержимым

Содержимое элемента получаем через innerHTML
(предпочтительно) или innerText

```
let strValue = element.innerHTML  
element.innerHTML = strValue + " и даже больше"
```

Манипуляции с атрибутами

Большинство атрибутов имеют аналоги в свойствах элемента, также их можно получить через метод `el.getAttribute(attribute)` и установить через `el.setAttribute(attribute, value)`

```
let image = document.createElement('img')

let flag = checkboxElement.checked
let text = inputElement.value
let selected = selectElement.selectedOptions[0].value
image.src = './images/balzhan.jpg'
// или
image.setAttribute('img', './images/balzhan.jpg')

document.body.appendChild(image)
```


Манипуляции со стилями и классами

```
// установка стилей
element.style.color = "#565656"
element.style.borderBottom = "1px solid #ddd"
element.style.transition = ".5s"

element.classList.add("py-3")           //добавить класс
element.classList.remove("mb-5")        //удалить класс
console.log(element.classList.contains("text-center")) //проверить наличие
setInterval(() => element.classList.toggle("fade"), 1000) //переключить класс
```

Обработчики событий

Событие – это сигнал от браузера о том, что что-то произошло. Все DOM-узлы подают такие сигналы (хотя события бывают и не только в DOM).

Вот список самых часто используемых DOM-событий, пока просто для ознакомления:

<https://learn.javascript.ru/introduction-browser-events>

Навешивание обработчика клика на кнопку (пример)

```
<body>
  <div class="container py-5">
    <button class="btn btn-primary" id="toggler">Нажми</button>
  </div>
  <script>

    const btn = document.getElementById('toggler')

    btn.onclick = e => {
      | alert('Ай')
    }
    //или так
    btn.onclick = function(e) {
      | alert('Ай')
    }
    //или так
    btn.addEventListener('click', e => {
      | alert('Ай')
    })

  </script>
```

Аргумент e – это объект, содержащий детали события, например, элемент, на котором оно произошло, координаты и т.д.

Подробнее можно смотреть, логируя его через `console.log(e)`

Объект события Event

```
▼ click { target: button#toggler.btn.btn-lg.w-100.btn-primary , buttons: 0, clientX: 231, clientY: 170, layerX: 231, task1.html:36:21  
  layerY: 170 }  
    altKey: false  
    bubbles: true  
    button: 0  
    buttons: 0  
    cancelBubble: false  
    cancelable: true  
    clientX: 231  
    clientY: 170  
    composed: true  
    ctrlKey: false  
    ▶ currentTarget: <button id="toggler" class="btn btn-lg w-100 btn-primary">  
    defaultPrevented: false  
    detail: 1  
    eventPhase: 2  
    ▶ explicitOriginalTarget: <button id="toggler" class="btn btn-lg w-100 btn-primary">  
    isTrusted: true  
    layerX: 231  
    layerY: 170  
    metaKey: false  
    movementX: 0  
    movementY: 0  
    mozInputSource: 1  
    mozPressure: 0  
    offsetX: 218  
    offsetY: 37  
    ▶ originalTarget: <button id="toggler" class="btn btn-lg w-100 btn-primary">  
    pageX: 231  
    pageY: 170  
    rangeOffset: 3  
    ▶ rangeParent: #text "Я нажат"  
    region: ""  
    relatedTarget: null  
    returnValue: true  
    screenX: 231  
    screenY: 255  
    shiftKey: false
```

Что такое event.preventDefault()?

Нужно, чтобы предотвратить стандартное поведение элемента и вместо этого выполнить свой обработчик

Чаще всего используется на событии отправки формы или нажатия ссылки

<https://developer.mozilla.org/ru/docs/Web/API/Event/preventDefault>

```
const form = document.getElementById('eq-form')

form.onsubmit = e => {
  e.preventDefault()
  console.log(e)
}
```

Удобное обращение к элементам формы

```
const form = document.getElementById('eq-form')  
  
let elements = form.elements // просто список всех элементов  
  
let a = form.elements.a // input name="a"  
let b = form.elements.b // input name="b"  
let c = form.elements.c // input name="c"
```

Полезные ссылки

<https://learn.javascript.ru/array-methods>

<https://developer.mozilla.org/ru/docs/Web/API/Event>

<https://developer.mozilla.org/ru/docs/Web/API/Event/preventDefault>

<https://learn.javascript.ru/form-elements>

<https://learn.javascript.ru/dom-nodes>

<https://learn.javascript.ru/datetime>