



ERASMUS+

Enriching lives, opening minds

WEB-TECHNOLOGIES

JavaScript ES6 event-driven programming

YAROSHEVICH
Andrey Olegovich

www.YAROSHEVICH.com

ao@asp.net.by

+375 29 254-07-92

Web Technologies with C# and JavaScript.

The process of client side scripting



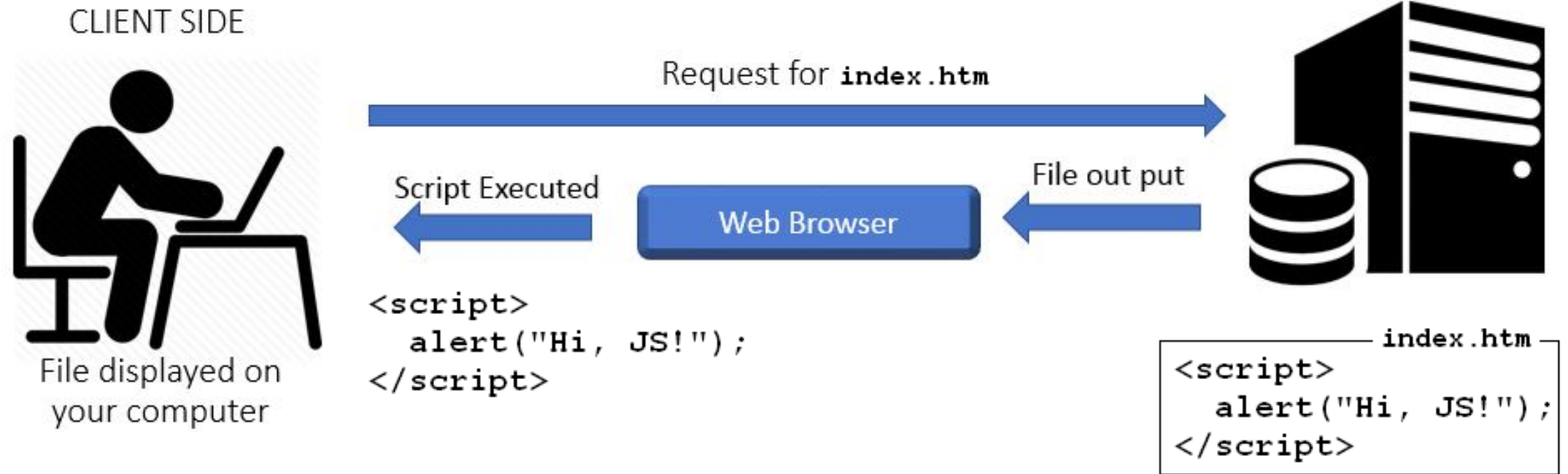
Back End

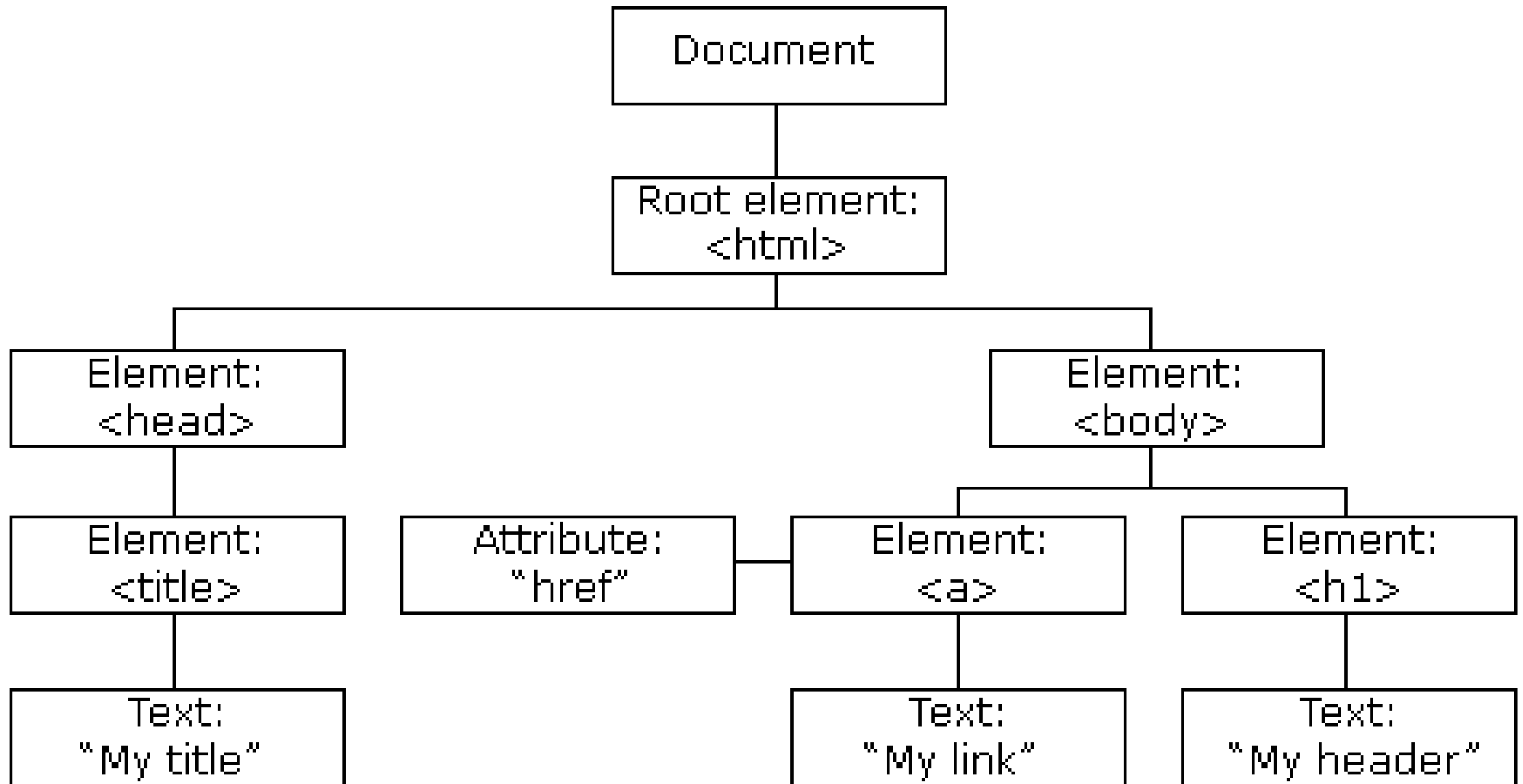
The process of server side scripting



FrontEnd

The process of client side scripting





HTML Page Structure

`<!DOCTYPE html>` ← Tells version of HTML

`<html>` ← HTML Root Element

`<head>` ← Used to contain page HTML metadata

`<title>Page Title</title>` ← Title of HTML page

`</head>`

`<body>` ← Hold content of HTML

`<h2>Heading Content</h2>` ← HTML heading tag

`<p>Paragraph Content</p>` ← HTML paragraph tag

`</body>`

`</html>`

```
<script>
  switch(new Date().getDay()){
    case 3:
      text="Wed";
      break;
    default:
      text="Other Day";
  }
  alert(text);
</script>
```

HTML Page Structure

`<!DOCTYPE html>` ← Tells version of HTML

`<html>` ← HTML Root Element

`<head>` ← Used to contain page HTML metadata

`<title>Page Title</title>` ← Title of HTML page

`</head>`

`<body>` ← Hold content of HTML

`<h2>Heading Content</h2>` ← HTML heading tag

`<p>Paragraph Content</p>` ← HTML paragraph tag

`</body>`

`</html>`

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>WEB-technologies</title>

    <script>
      switch(new Date().getDay()){
        case 3:
          text="Wed";
          break;
        default:
          text="Other Day";
        }
      alert(text);
    </script>
  </head>
```

CLIENT SIDE



File displayed on
your computer



Script Executed



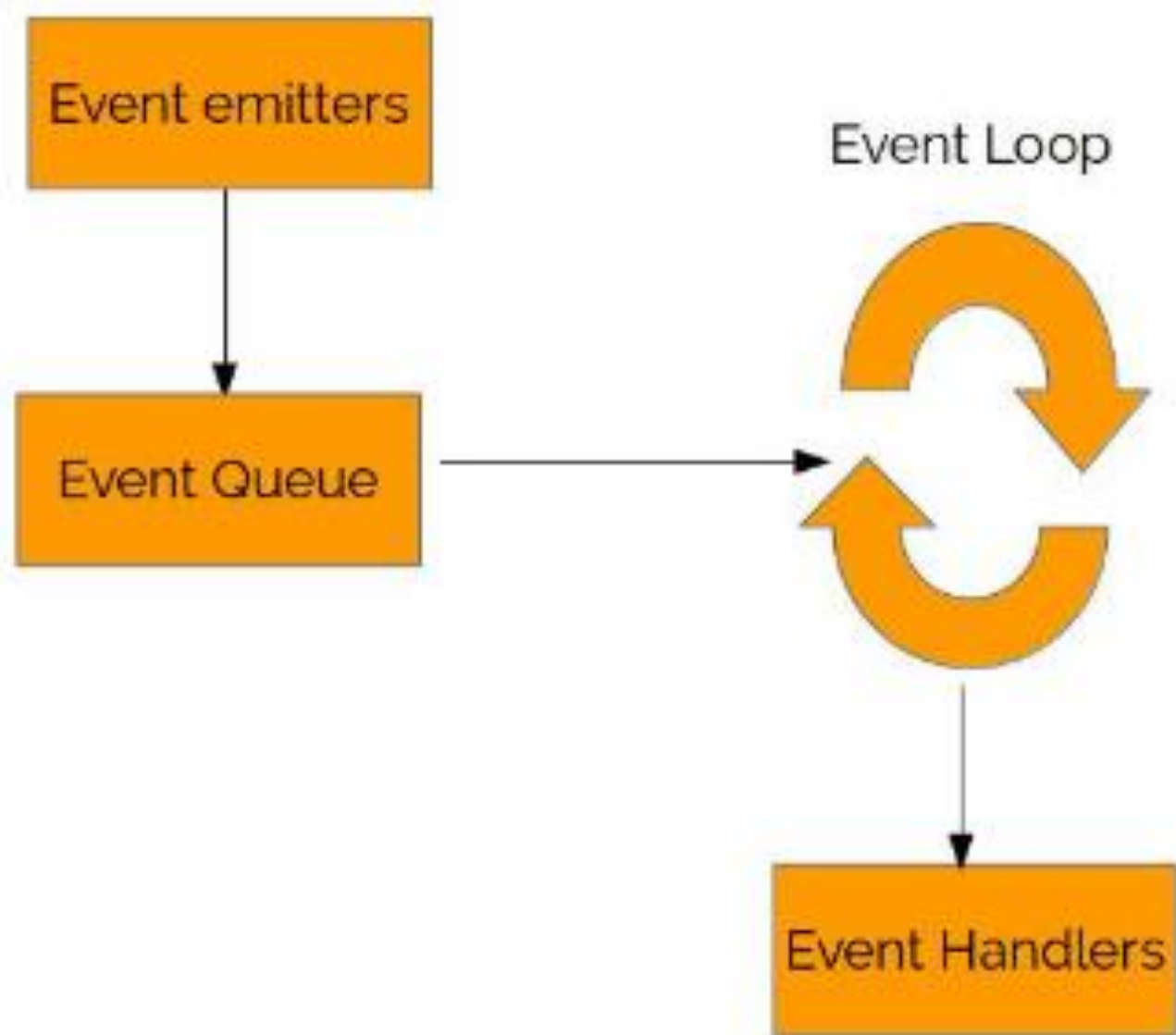
Web Browser

```
<script>  
  alert("Hi, JS!");  
</script>
```

single quotation marks



double quotation marks



What are some examples of events?



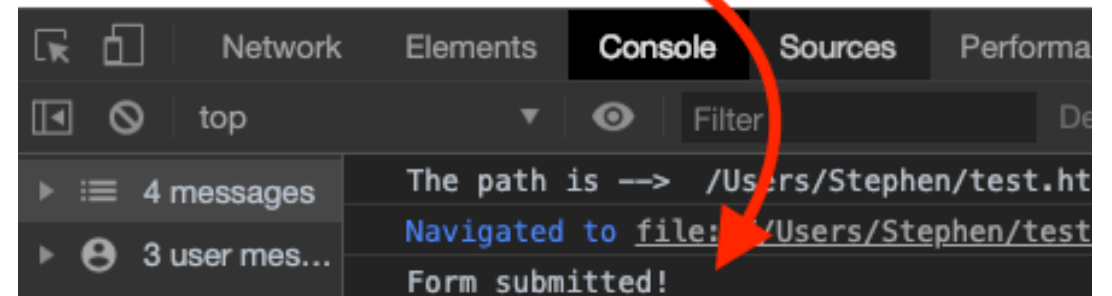
Select Option ▾

Select Option

- Option 1
- Option 2
- Option 3
- Option 4
- Option 5

Stephen Submit

Form submitted;
entry still visible.



Ways of using JavaScript Events

```
graph TD; A[Ways of using JavaScript Events] --> B[Inline/HTML Event handlers]; A --> C[addEventListener()]; A --> D[Traditional DOM Event handler];
```

**Inline/HTML
Event handlers**

addEventListener()

**Traditional DOM
Event handler**

```
<head>
```

```
  <title>Hello</title>
```

```
  <script>
```

```
    function Hello(){
```

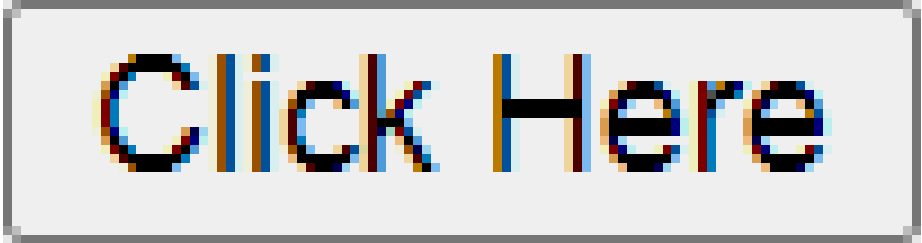
```
      alert("Hello");
```

```
    }
```

```
  </script>
```

```
</head>
```

```
<button> Click Here</button>
```

A rectangular button with a light gray background and a thin dark border. The text "Click Here" is centered on the button in a pixelated, multi-colored font. The letters are primarily black with blue, red, and yellow outlines, giving it a retro, digital appearance.

Click Here

```
<head>
  <title>Hello</title>
  <script>
    function Hello(){
      alert("Hello");
    }
  </script>
</head>
<body>
  <button onclick="Hello()">Click Here</button>
```

Hello JavaScript!

A rectangular button with rounded corners, a light gray background, and a thin black border. The text "Click Me!" is centered on the button in a black, sans-serif font.

Click Me!

querySelector

First we need to access the button element in our JavaScript code.

*variable name. We
store the button in
this variable to
reuse it later*

*selects the first
HTML element
which the query
applies to*

```
const button = document.querySelector("button");
```

*variable definition,
we could use let or
var instead. But
const is preferred*

*The context in which
the querySelector will
search for the given
query. Here this is the
HTML file*

*same syntax as CSS
selectors. you can
refer to elements,
ids or classes here.*

```
<head>
  <title>Hello</title>
  <script>
    function Hello(){
      document.querySelector('h2').innerHTML=
        "Hello JavaScript!";
    }
  </script>
</head>
<body>
  <h2></h2>
  <button onclick="Hello()">Click Me!</button>
```

Hello JavaScript!

Click Me!

```
document.querySelector('button').onclick = Hello;
```

```
<head>
```

```
  <title>Hello</title>
```

```
  <script>
```

```
    function Hello(){
```

```
      document.querySelector('h2').innerHTML=
```

```
      "Hello JavaScript!";
```

```
      document.querySelector('button').onclick = Hello;
```

```
    }
```

```
  </script>
```

```
</head>
```

```
<body>
```

```
<h2></h2>
```

```
<button onclick="Hello()">Click Me!</button>
```

Hello JavaScript!

Click Me!



Counter



https://bip1252.w3spaces.com/counter.html

0

Count



Инспектор



Консоль



Отладчик



Сеть



Стили



Профайлер



Память



Поиск в консоли



Uncaught TypeError: document.querySelector(...) is null
<anonymous> https://bip1252.w3spaces.com/counter.html:16
[\[Подробнее\]](#)



<https://web-t.w3spaces.com/>

[0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#)

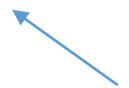
WEB-T by Yevgeniya Subotka



[Click Here](#)

```
<head>
  <title>Hello</title>
  <script>
    function Hello(){
      document.querySelector('h2').innerHTML=
        "Hello JavaScript!";
      document.querySelector('button').onclick = Hello;
    }
  </script>
</head>
```

going to try and find a button on line 18



```
</head>
```

```
<body>
```

but the button doesn't show up in my HTML page yet

```
<h2></h2>
```

```
<button>Click Me!</button>
```



```
<button>Count</button>
```

```
<script>
```

```
  document.querySelector('button').onclick = Hello;
```

```
</script>
```

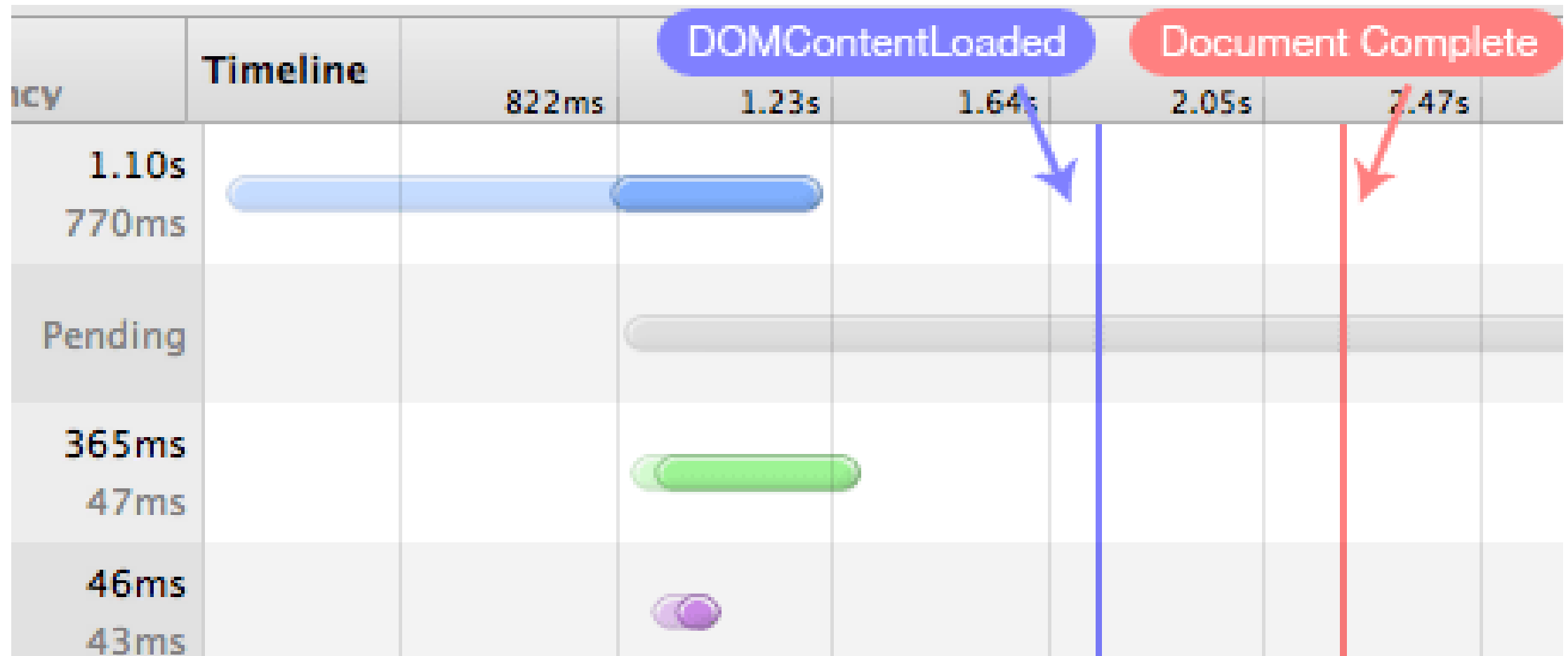
addEventListener

```
document.addEventListener('DOMContentLoaded',  
    function()  
    {  
        // Some code here  
    }  
);
```

DOMContentLoaded

DOMContentLoaded event is fired,
the page's DOM is parsed and ready.

Document Complete is actually
the point in time when **all the
content referenced in the
HTML is fully-loaded.**



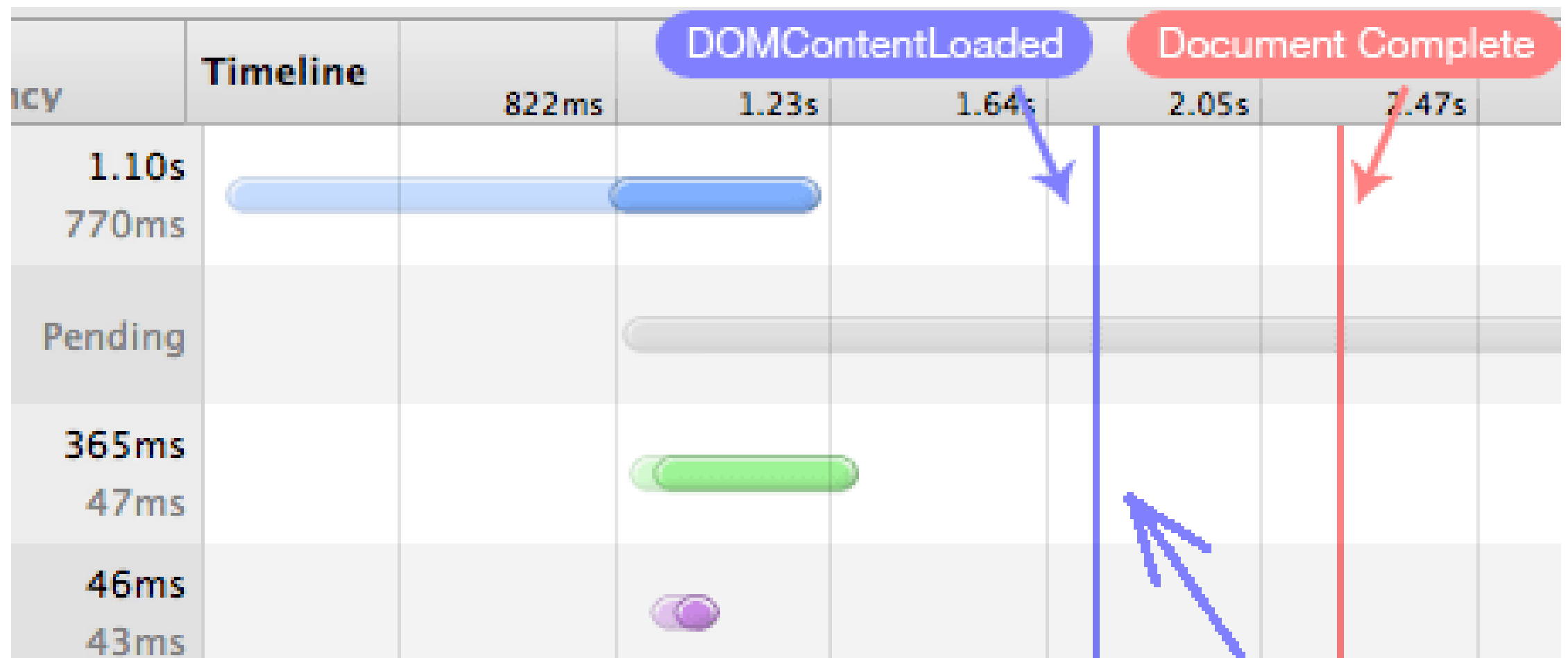
```
document.addEventListener('DOMContentLoaded',function(){})
```

```
document.addEventListener('DOMContentLoaded',  
    function() ← anonymous function  
    {  
        // Some code here, which should run  
        // when the DOM is done loading  
    }  
);
```

```
function Hello(){  
    alert("Hello!");  
}  
document.addEventListener('DOMContentLoaded',function(){  
    document.querySelector('button').onclick = Hello;  
});
```

```
document.addEventListener('DOMContentLoaded',function(){
    document.querySelector('button').addEventListener('click',Hello);
});

document.addEventListener('DOMContentLoaded',function(){
    document.querySelector('button').onclick = Hello;
});
```



```
function(){  
    document.querySelector('button').onclick = Hello;  
}
```



External Javascript

```
<head>  
  <script type="text/javascript"  
    src="external_js.js"></script>  
</head>
```

+ JavaScript libraries

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" ></script>
```

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"></script>
```

```
<script>  
    $(document).ready(function () {  
        $('[data-toggle="tt"]').tooltip();  
    });  
</script>
```

WEB-technologies
introduction

0. W-T Intro 1.

```
<a href="Projects/0/" data-toggle="tt" title="WEB-technologies  
introduction">0. W-T Intro</a>
```

<https://web-t.w3spaces.com/hello.html>

```
<script>
```

```
function hello() {
```

```
    const heading = document.querySelector('h1');
```

```
    if (heading.innerHTML === 'Hello!') {
```

```
        heading.innerHTML = 'Goodbye!';
```

```
    } else {
```

```
        heading.innerHTML = 'Hello!';
```

```
    }
```

```
}
```

```
</script>
```

HTML forms

The placeholder - the thing the user sees filled into that input field originally

automatically focus this input field

```
<form>  
  <input autofocus id="name" placeholder="Name" type="text">  
  <input type="submit">  
</form>
```

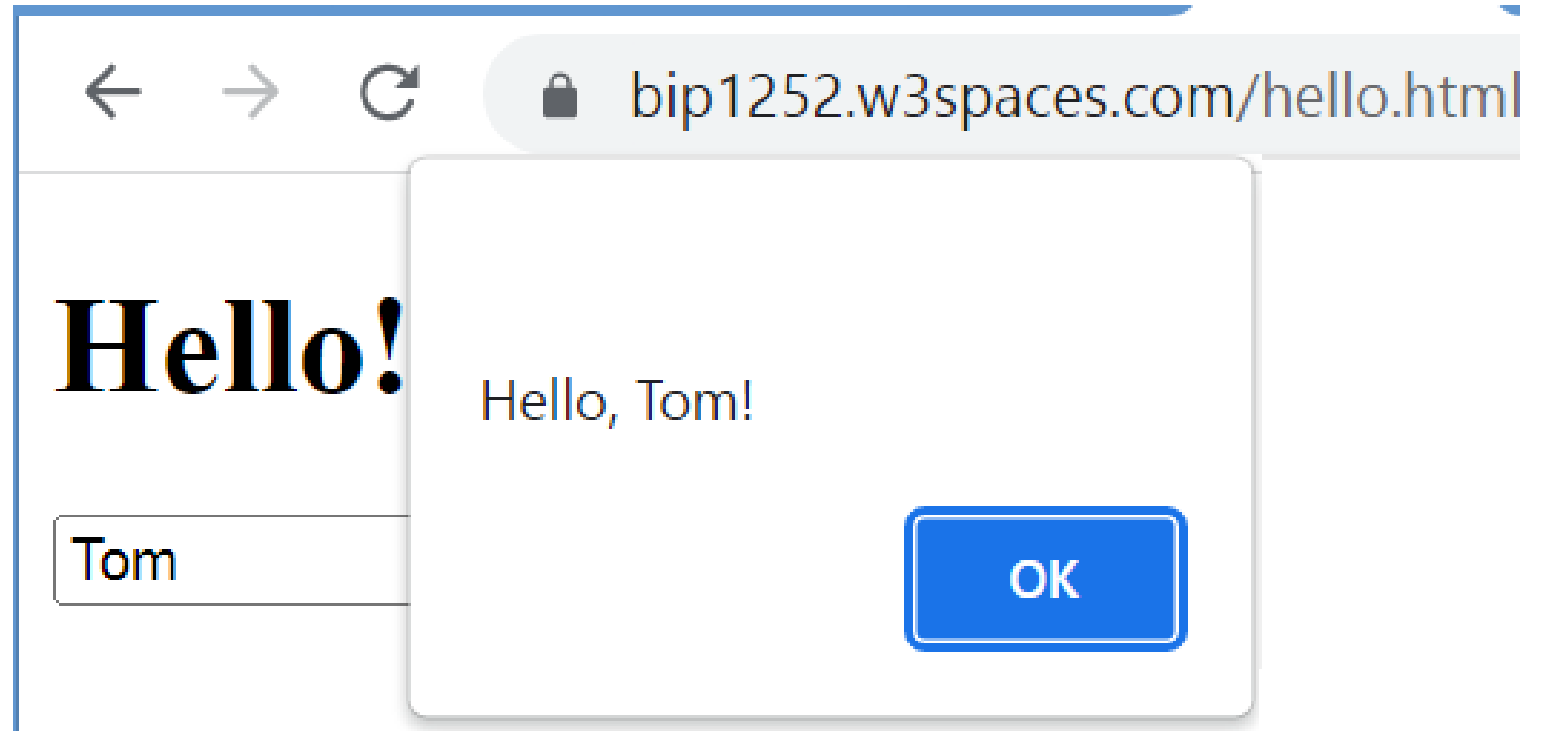
Visual representation of the HTML form:

Input field: Name

Submit button: Отправить

input field where the user can type in some text

```
document.addEventListener('DOMContentLoaded'  
, function() {  
.....  
});
```



```
document.querySelector(' form ').
```

```
<body>
```

```
  <h1>Hello!</h1>
```

```
  <form>
```

```
    <input autofocus id="name" placeholder="Name" type="text">
```

```
    <input type="submit">
```

```
  </form>
```

```
</body>
```

```
document.querySelector('form').onsubmit=f
```

```
function f() {
```

```
}
```

Anonymous function

```
<script>
```

```
document.addEventListener('DOMContentLoaded', function() {  
    document.querySelector('form').onsubmit =
```

```
function() {  
    let name = document.querySelector('#name').value;  
    alert(`Hello, ${name}!`);  
};
```

```
});
```

```
</script>
```

.blue

{

color:blue;

}

CSS Class
definition
starts with "."

document.querySelector('tag')
document.querySelector('#id')
document.querySelector('.class')

This tells which class to
use to style

<h1 class="blue">Center-aligned heading</h1>

CSS Class

JavaScript

```
const element =  
document.querySelector(".blue");  
element.innerHTML = "WEB-T";
```

```
<form>
```

```
  <input autofocus id="name" placeholder="Name" type="text">
```

```
  <input type="submit">
```

```
</form>
```

```
<script>
```

```
  document.addEventListener('DOMContentLoaded', function() {
```

```
    document.querySelector('form').onsubmit = function() {
```

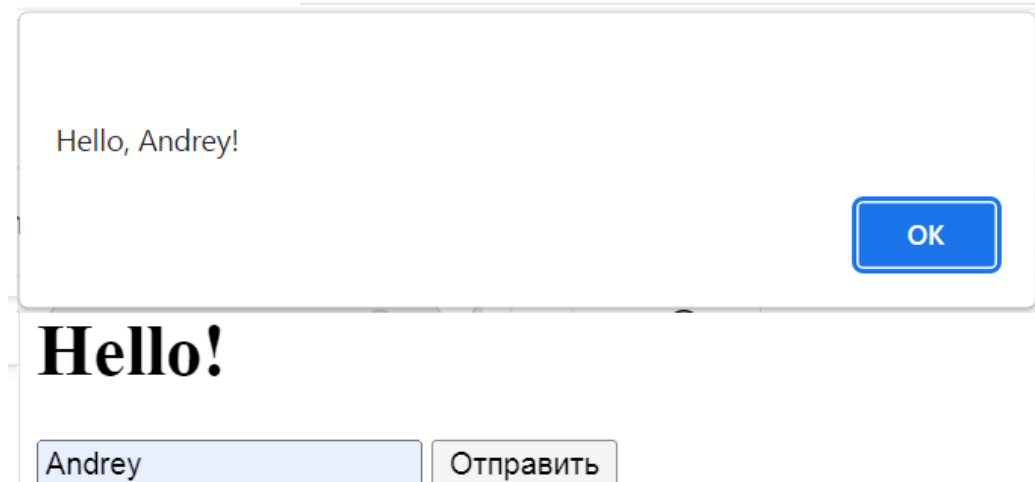
```
      let name = document.querySelector('#name').value;
```

```
      alert(`Hello, ${name}!`);
```

```
    };
```

```
  });
```

```
</script>
```



The image shows a screenshot of a web application. At the top, there is a form with a text input field containing the name "Andrey" and a submit button labeled "Отправить" (Submit). Below the form, an alert dialog box is displayed with the message "Hello, Andrey!" and an "OK" button. The text "Hello!" is also visible below the alert dialog.

change CSS

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Colors</title>
5    </head>
6    <body>
7      <h1 id="hello">Hello!</h1>
8      <button>Red</button>
9      <button>Blue</button>
10     <button>Green</button>
11   </body>
12 </html>
```

Hello!

RedBlueGreen

```
<script>
```

```
document.addEventListener('DOMContentLoaded',
```

```
    function() {
```

```
    };
```

```
});
```

```
</script>
```

```
<button id="red">Red</button>  
<button id="blue">Blue</button>  
<button id="green">Green</button>
```

```
).style.color = 'red';
```

Hello!

```
// Change font color to red
```

Red

Blue

Green

```
document.querySelector('#red').onclick =
```

```
function() {
```

```
document.querySelector('#hello').style.color = 'red';  
};
```

```
<h1 id="hello">Hello!</h1>
```

<https://web-t.w3spaces.com/colors.html>

Hello!

// Change font color to blue

```
document.querySelector('#blue').onclick = function() {  
    document.querySelector('#hello').style.color = 'blue';  
};
```

// Change font color to green

```
document.querySelector('#green').onclick =function() {  
    document.querySelector('#hello').style.color = 'green';  
};
```

Hello!

same code

Hello!

// Change font color to blue

Red Blue Green

```
document.querySelector('#blue').onclick = function() {  
    document.querySelector('#hello').style.color = 'blue';  
};
```

// Change font color to green

```
document.querySelector('#green').onclick =function() {  
    document.querySelector('#hello').style.color = 'green';  
};
```

Hello!

Red Blue Green

Data attribute - **data-color**

```
<button data-color="red">Red</button>
```

```
<button data-color="blue">Blue</button>
```

```
<button data-color="green">Green</button>
```

querySelector

First we need to access the button element in our JavaScript code.

*variable name. We
store the button in
this variable to
reuse it later*

*selects the first
HTML element
which the query
applies to*

const button = document.querySelector("button");

*variable definition,
we could use let or
var instead. But
const is preferred*

*The context in which
the querySelector will
search for the given
query. Here this is the
HTML file*

*same syntax as CSS
selectors. you can
refer to elements,
ids or classes here.*



querySelector() ⇒ 

querySelectorAll() ⇒ [, ]

`document.querySelectorAll('button')`

https://bip1252.w3spaces.com/color.html

Инспектор **Консоль** Отладчик Сеть Стили Профайлер

Поиск в консоли

▶ GET <https://bip1252.w3spaces.com/favicon.ico>

>> document.querySelector('button');

← ▶ <button id="red">

>> document.querySelectorAll('button');

← ▼ NodeList(3) [button#red, button#blue, button#green]

▶ 0: <button id="red">

▶ 1: <button id="blue">

▶ 2: <button id="green">

length: 3

▶ <prototype>: NodeListPrototype { item: item(), keys: keys(), values: values(), ... }

>>



querySelector()



```
>> document.querySelector('button');
```

```
<button id="red"> ❌
```



querySelectorAll() ⇒ [, ]
NodeList

```
>> document.querySelectorAll('button');
```

```
← ▼ NodeList(3) [ button#red , button#blue , button#green ]
```

```
  ▶ 0: <button id="red">
```

```
  ▶ 1: <button id="blue">
```

```
  ▶ 2: <button id="green">
```

```
    length: 3
```

```
  ▶ <prototype>: NodeListPrototype { item: item(), keys: keys(), values: values(), ... }
```

```
>> const forbes = ["Arnault", "Musk", "Bezos"];
```

```
← undefined
```

```
>> forbes[0]
```

```
← "Arnault"
```

```
>> forbes.length
```

```
← 3
```

Red

Blue

Green

Event Handler

Event Handler

Event Handler

forEach



Сорока-Белобока кашу варила,
Деток кормила: - Этому дала, - Этому дала,

[ , ].forEach();

```
document.querySelectorAll('button').forEach(function(button) {  
    button.onclick = function() {  
        document.querySelector('#hello').style.color =  
            button.dataset.color;  
    };  
});
```

Hello!

Red

Blue

Green

When the page is done loading

<script>

document.addEventListener('DOMContentLoaded', function() {

I'm going to document.querySelectorAll,
looking for all of the buttons

document.querySelectorAll('.color-change').forEach(

function(button) {

looked for things of the particular class

button.onclick = function() {

document.querySelector('#hello').style.color = button.dataset.color;
};

});

});

</script>

<button class="color-change" data-color="red">Red</button>

<button class="color-change" data-color="blue">Blue</button>

<button class="color-change" data-color="green">Green</button>

function that takes as input,
the button

It's first going to pass in as
input the first button

Red

then the second button

Blue

then the third button

Green

When the button is clicked on, go
ahead and run this function

function(but) {

but.onclick = function() {

get me the #hello element

change its color to **but.dataset.color**

document.querySelector('#hello').style.color =

but.dataset.color;

};

<button class="color-change" data-color="red">Red</button>

```
>> document.querySelectorAll('button');
```

```
← ▾ NodeList(3) [ button#red , button#blue , button#green ]
```

```
▶ 0: <button id="red">
```

```
▶ 1: <button id="blue">
```

```
▶ 2: <button id="green">
```

```
length: 3
```

```
▶ <prototype>: NodeListPrototype { item: item(), keys: keys(), values: values(), ... }
```



bip1252.w3spaces.com/color.html



Bye

Red

Blue

Green



Элементы

Консоль

Источники

Сеть



top ▾



console.log(document.querySelecto



Все уровни ▾



Нет проблем

```
> let x= document.querySelector('#hello');
```

```
< undefined
```

```
> x.innerHTML
```

```
< 'Hello!'
```

```
> x.innerHTML="Bye";
```

```
< 'Bye'
```

```
> x.style.color="yellow";
```

```
< 'yellow'
```

```
> x.style.color="darkgray";
```

```
< 'darkgray'
```

function() {} () => {}

<pre>function() { document.querySelectorAll('.color- change').forEach(function(but) { but.onclick = function() {document.querySelector('#hello').style.color = but.dataset.color; }; }); });</pre>	<pre>() => { document.querySelectorAll('.color- change').forEach(but => { but.onclick = () => { document.querySelector('#hello').style.color = but.dataset.color; }; }); });</pre>
--	--

function() {...}

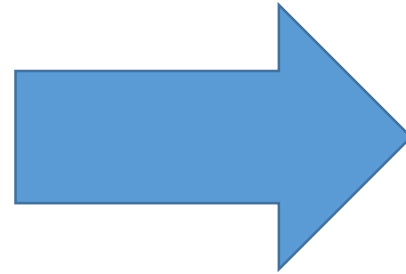
() => {...}

function(but) {
but.onclick =
function() {...}
}

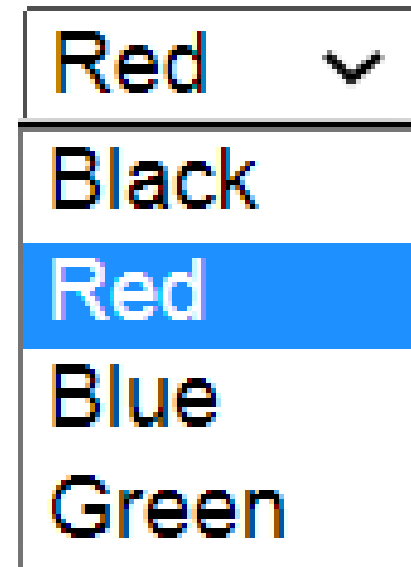
but => {
but.onclick =
() => {...}
}

Bunch of buttons vs Dropdown

Hello!



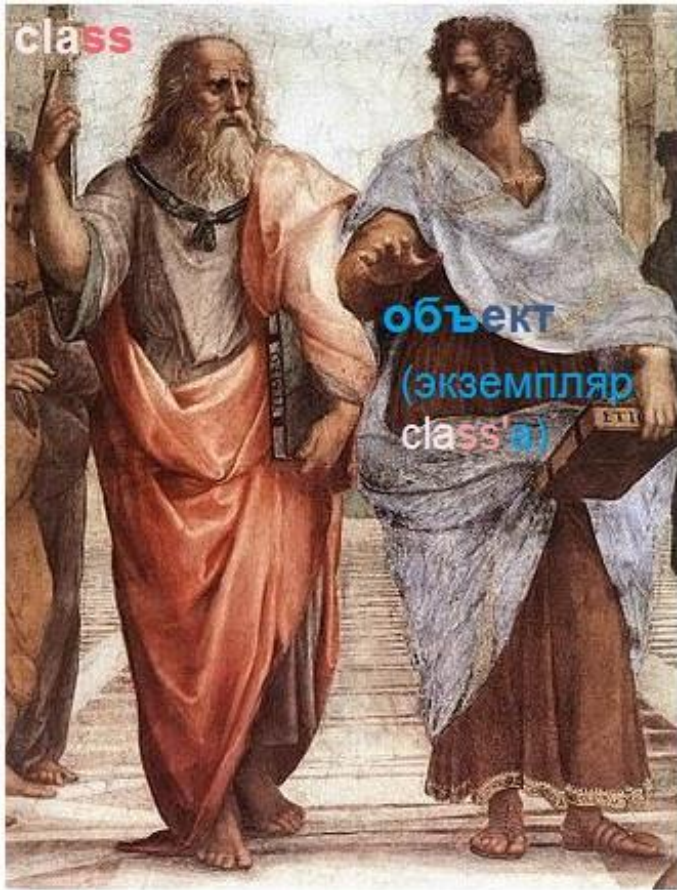
Hello!



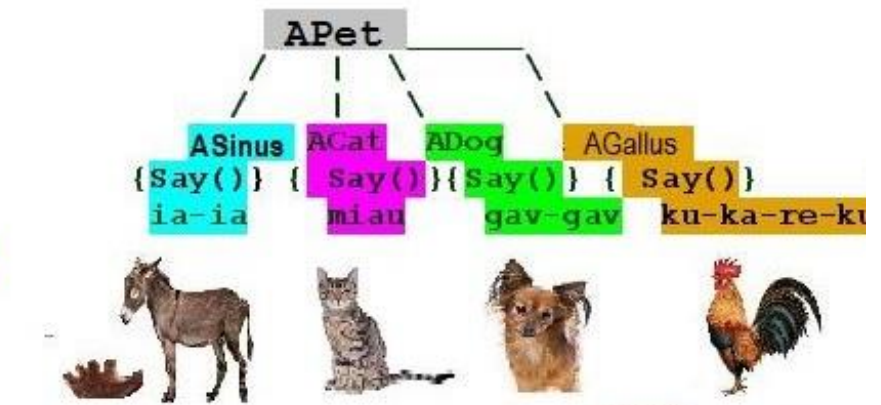
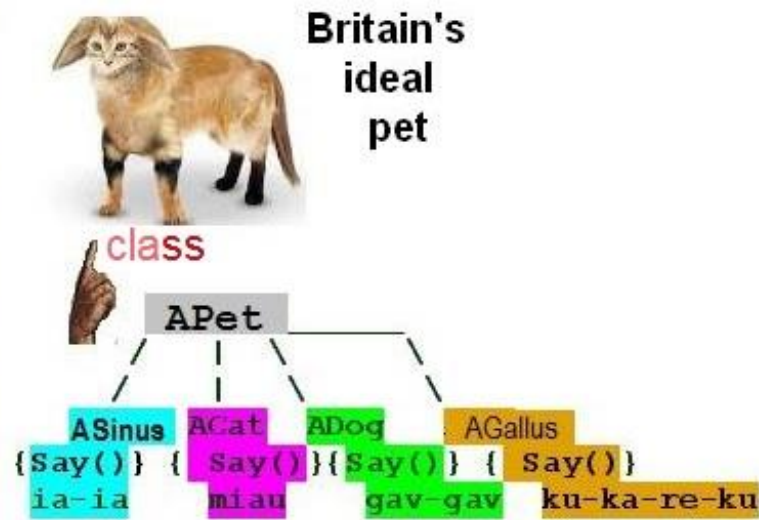
```
<head>
  <script>
    document.addEventListener('DOMContentLoaded', () => {
      document.querySelector('#color-change').onchange = function() {
        document.querySelector('#hello').style.color = this.value;
      };
    });
  </script>
</head>
<body>
  <h1 id="hello">Hello!</h1>
  <select id="color-change">
    <option value="red">Red</option>
    <option value="blue">Blue</option>
    <option value="green">Green</option>
  </select>
</body>
```



Object-Oriented Programming



Amicus Plato, sed magis amica veritas
Plato is my friend, but Truth is a better friend



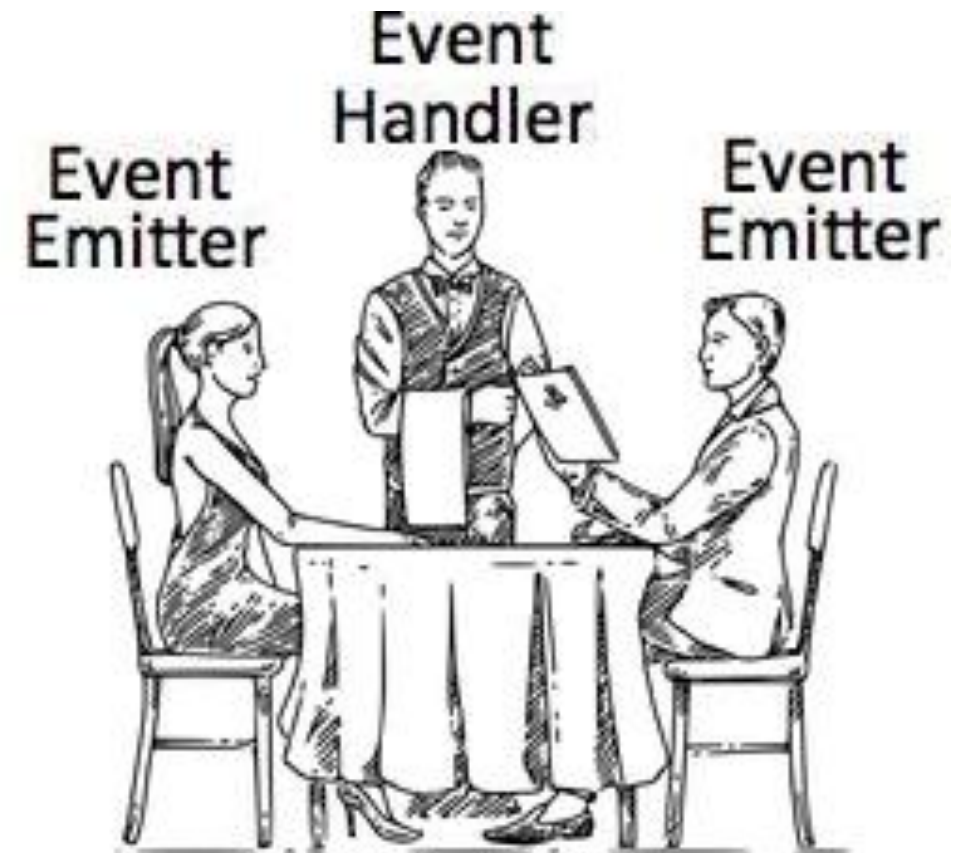
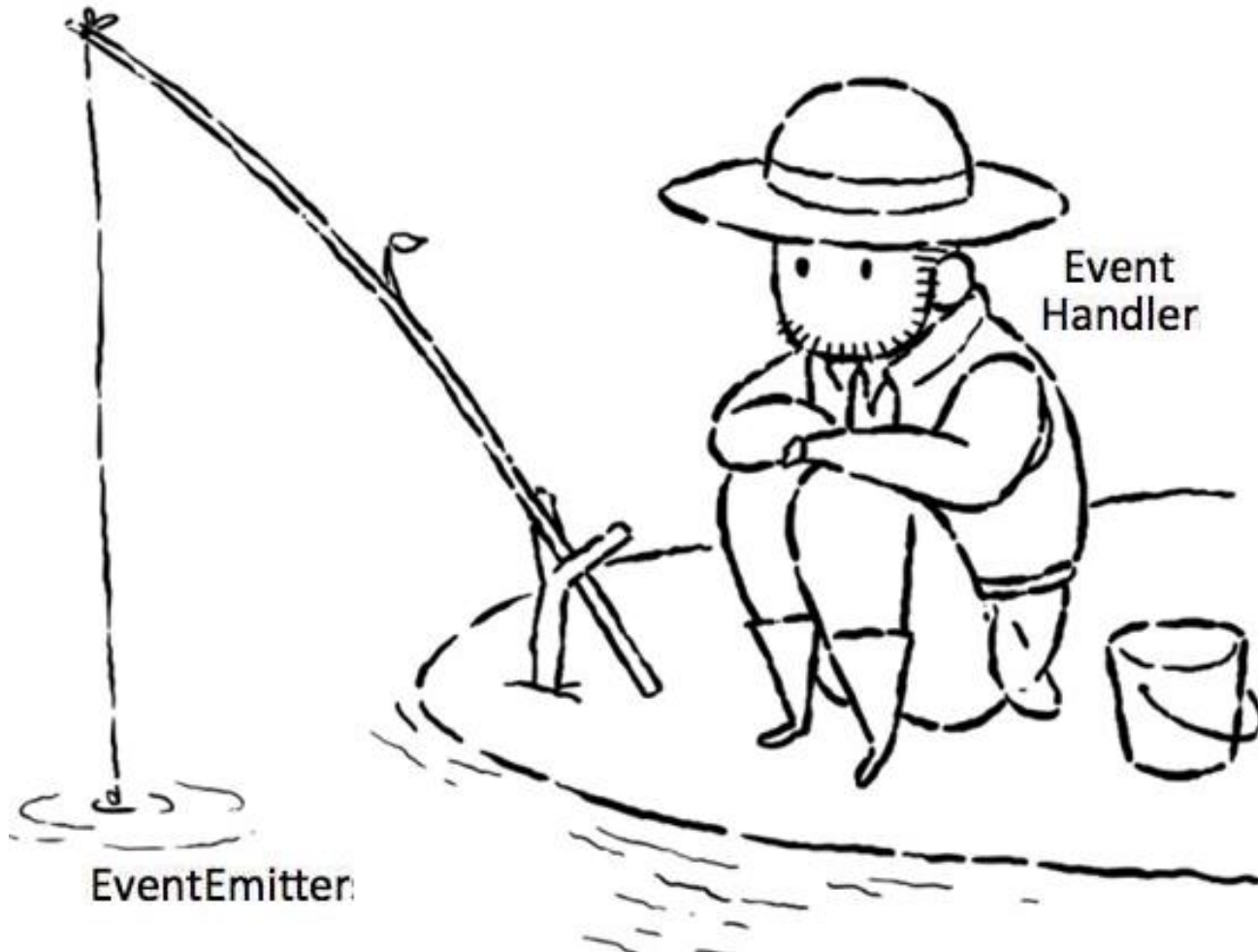
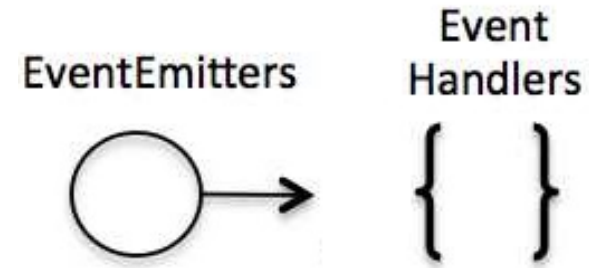
объект
(экземпляр class'a)

Л.Н. ТОЛСТОЙ

ЛЕВ
и
СОБАЧКА



Event Driven Programming



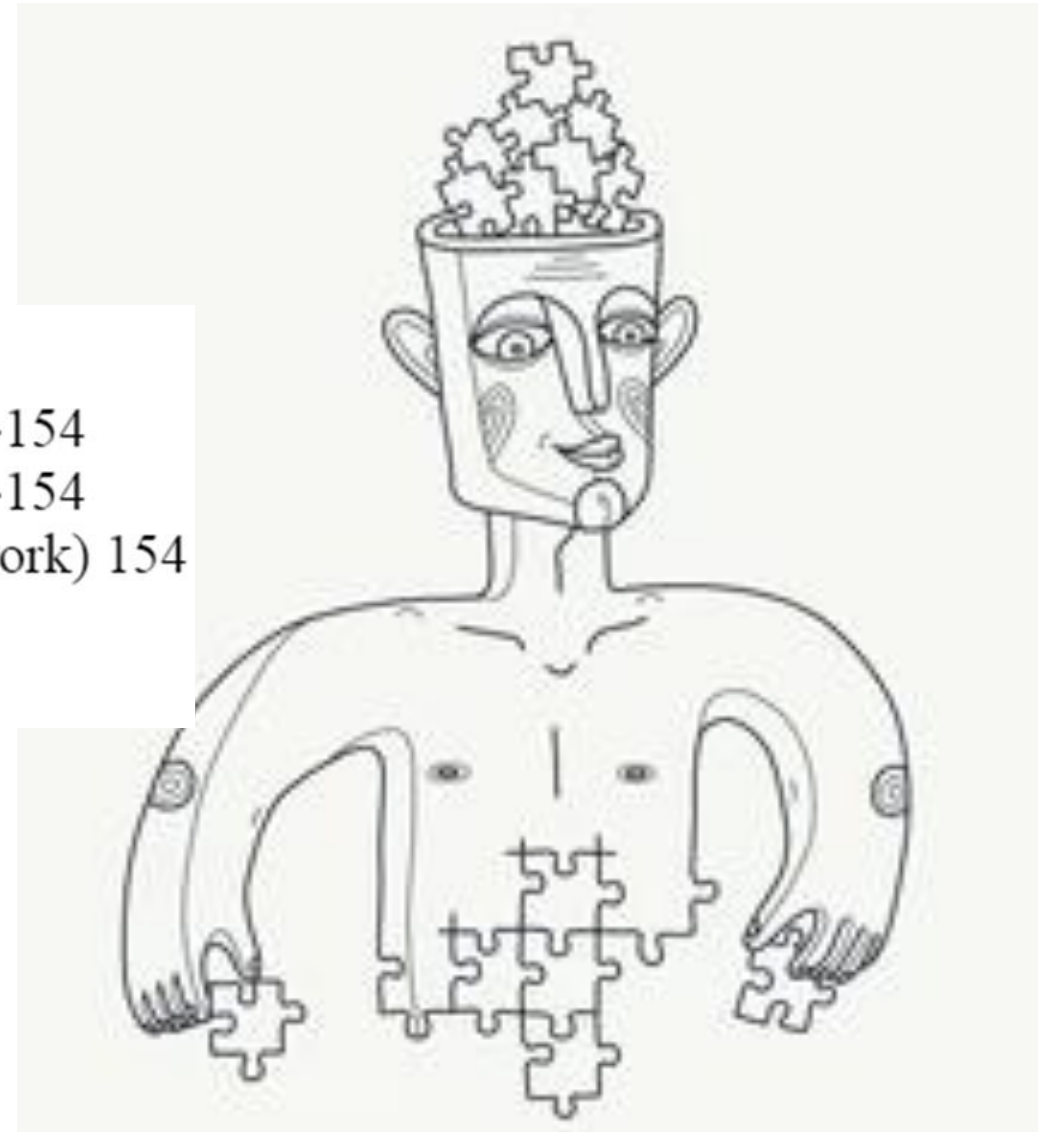
Self Made Man

To Do

- 13.11.2024 08:30 WEB technologies (Lecture) 153-154
- 13.11.2024 10:00 WEB technologies (Lecture) 153-154
- 14.11.2024 11:50 WEB technologies (Laboratory work) 154

14.11.2024 13:20 WEB technologies (Lab

Отправить



To Do

- 13.11.2024 08:30 WEB technologies (Lecture) 153-154
- 13.11.2024 10:00 WEB technologies (Lecture) 153-154
- 14.11.2024 11:50 WEB technologies (Laboratory work) 154

```
<h1>To Do</h1>
```

```
<ul id="tasks">
```

```
</ul>
```

```
<form>
```

```
  <input id="task" autocomplete="off" size="33"
    autofocus placeholder="New Task" type="text">
```

```
</form>
```

```

<script>
  document.addEventListener('DOMContentLoaded', () => {

    document.querySelector('form').onsubmit = () => {

      const li = document.createElement('li');
      li.innerHTML =
        document.querySelector('#task').value;
      document.querySelector('#tasks').append(li);
      document.querySelector('#task').value = '';
      return false;
    };
  });
</script>

```

```

<h1>To Do</h1>
<ul id="tasks"></ul>
  <form>
    <input id="task" autocomplete="off" size="33"
      autofocus placeholder="New Task" type="text">
    <input type="submit">
  </form>

```

To Do

- 13.11.2024 08:30 WEB technologies (Lecture) 153-154
- 13.11.2024 10:00 WEB technologies (Lecture) 153-154
- 14.11.2024 11:50 WEB technologies (Laboratory work) 154

14.11.2024 13:20 WEB technologies (Labo

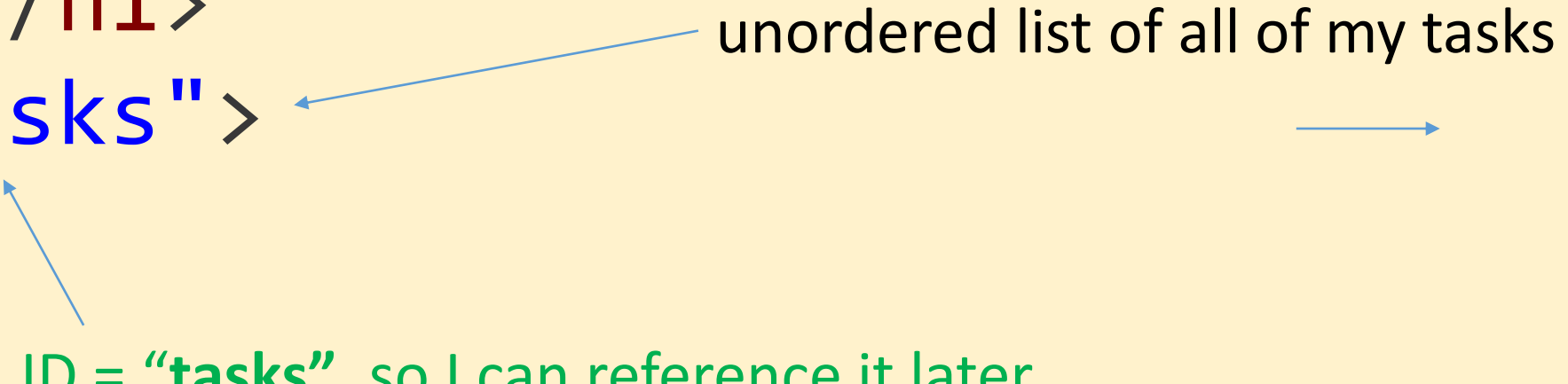
Отправить

```
<h1>To Do</h1>
```

```
<ul id="tasks">
```

```
</ul>
```

unordered list of all of my tasks



ID = "tasks", so I can reference it later

To Do

- 13.11.2024 08:30 WEB technologies (Lecture) 153-154

```
<h1>To Do</h1>
```

```
<ul id="tasks">
```

```
</ul>
```

```
<form>
```

```
  <input id="task"
    autocomplete="off"
    size="33"
```

```
    autofocus placeholder="New Task" type="text">
```

```
  <input type="submit">
```

```
</form>
```

input field & its an ID

input field - type is submit

To Do

Big heading

unordered list

New Task

text field where I can type in a task

Отправить

Submit button

Add <script> tag

```
<html lang="en">
```

```
  <head>
```

```
    <script>
```

```
      </script>
```

```
    <title>To Do</title>
```

```
  </head>
```

JavaScript to run after the DOM content is loaded

```
<script>  
  document.addEventListener('DOMContentLoaded', () =>  
  {  
    //JavaScript to run after the DOM content is loaded  
  
  });  
</script>
```

```
<script>
  document.addEventListener('DOMContentLoaded', () => {

    document.querySelector('form').onsubmit = () =>
    {

      };

    });
</script>
```



here's the function that I want to
run when the form is submitted

Add Task to List

```
<script>
  document.addEventListener('DOMContentLoaded', () => {
    document.querySelector('form').onsubmit = () => {
      // Create new item for list
      const li = document.createElement('li');
      li.innerHTML = document.querySelector('#task').value;

      // Add new item to task list
      document.querySelector('#tasks').append(li);

      // Clear input field
      document.querySelector('#task').value = '';
      // Stop form from submitting
      return false;
    };
  });
</script>
```

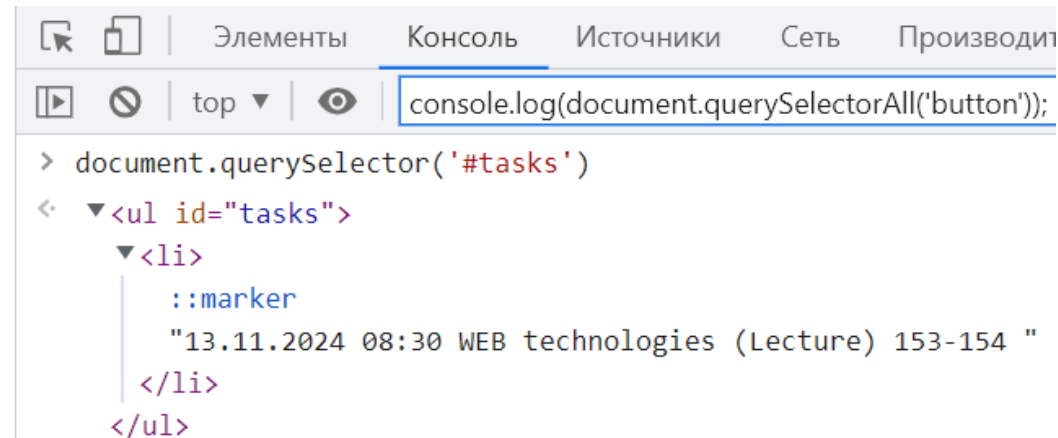
```
<script>
  document.addEventListener('DOMContentLoaded', () => {
    document.querySelector('form').onsubmit = () => {
      // Create new item for list
      const li = document.createElement('li');
      li.innerHTML = document.querySelector('#task').value;

      // Add new item to task list
      document.querySelector('#tasks').append(li);

      // Clear input field
      document.querySelector('#task').value = '';
      // Stop form from submitting
      return false;
    };
  });
```

To Do

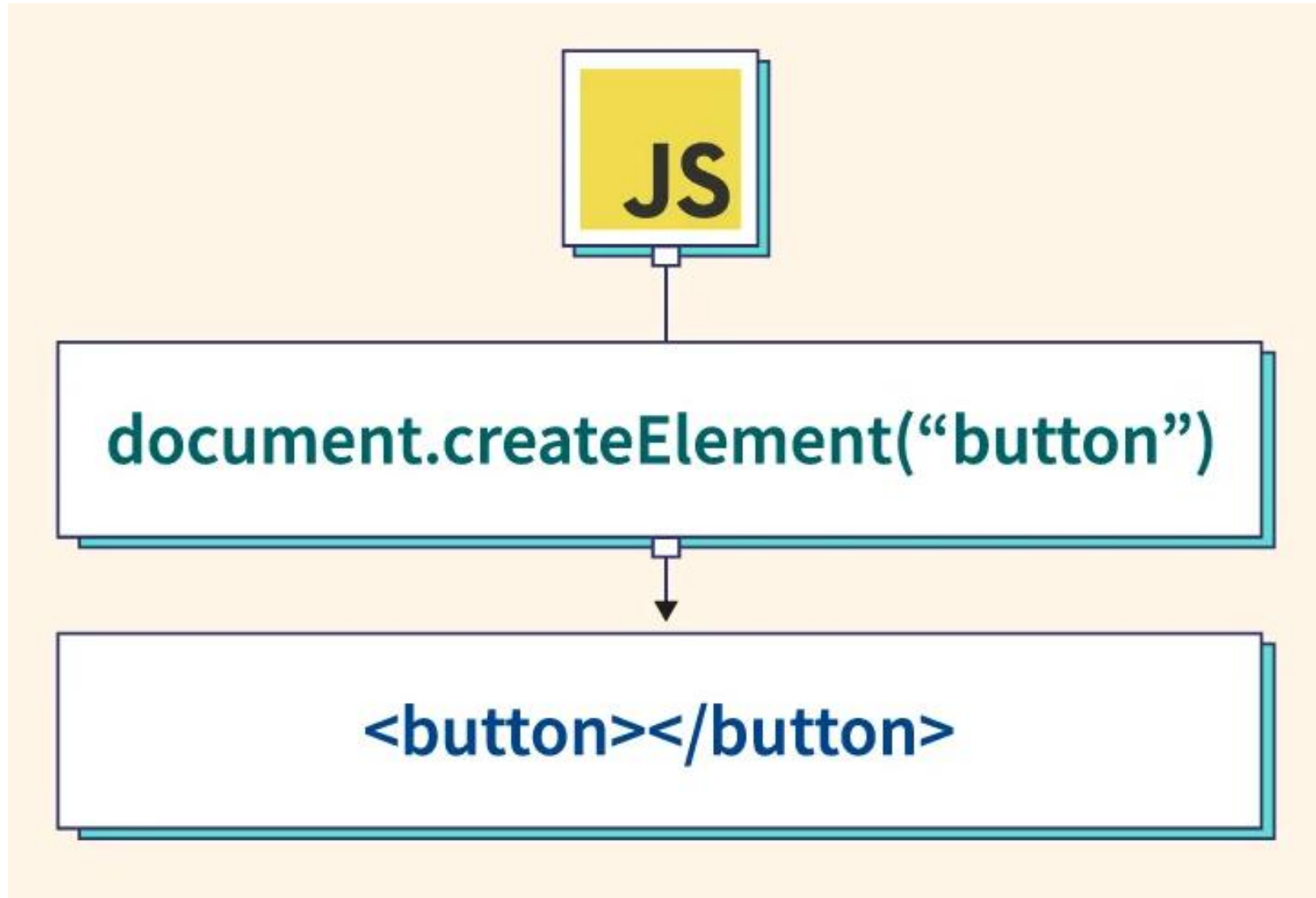
- 13.11.2024 08:30 WEB technologies (Lecture) 153-154



The screenshot shows a web browser's developer console with the 'Консоль' (Console) tab selected. The command `console.log(document.querySelectorAll('button'));` is entered in the input field. The output shows the result of `document.querySelector('#tasks')`, which is a `<ul id="tasks">` element. The console log displays the following structure:

```
> document.querySelector('#tasks')
< <ul id="tasks">
  <li>
    ::marker
    "13.11.2024 08:30 WEB technologies (Lecture) 153-154 "
  </li>
</ul>
```

document.createElement



Creating an element, a list item

```
const li = document.createElement('li');  
  
li.innerHTML =  
    document.querySelector('#task').value;
```

```
// Add new item to task list  
document.querySelector('#tasks').append(li);
```

```
<script>
  document.addEventListener('DOMContentLoaded', () => {
    document.querySelector('form').onsubmit = () => {
      // Create new item for list
      const li = document.createElement('li');
      li.innerHTML = document.querySelector('#task').value;

      // Add new item to task list
      document.querySelector('#tasks').append(li);

      // Clear input field
      document.querySelector('#task').value = '';
      // Stop form from submitting
      return false;
    };
  });
```

Submits the empty string as a task

To Do

-
-

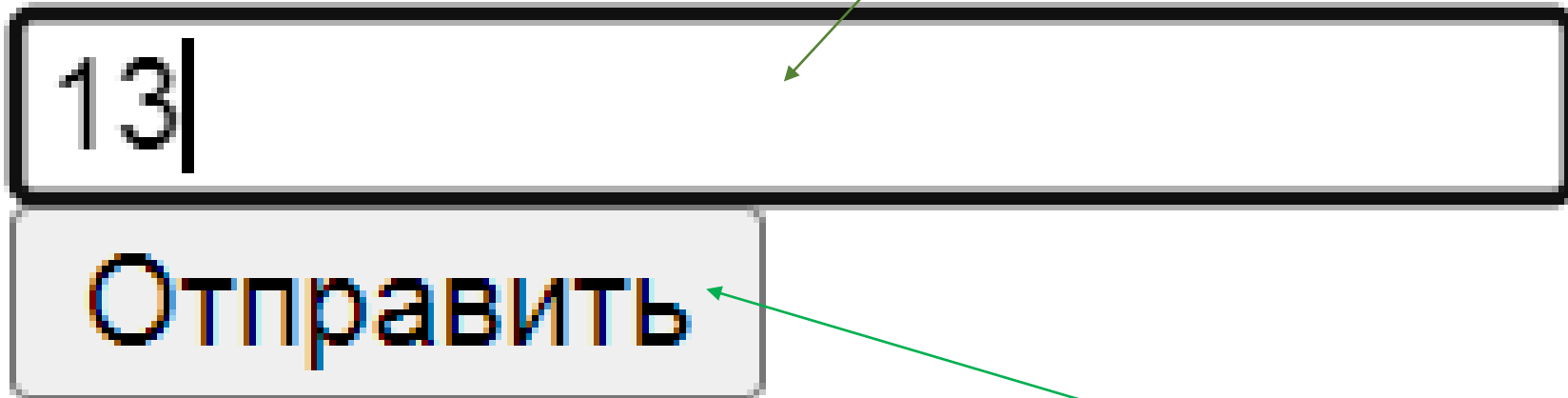
By default, submit button is disabled

Tasks

```
document.querySelector('#submit').disabled = true;
```

Tasks

as I begin to type things in



A web form consisting of a text input field and a submit button. The input field contains the text '13' and has a vertical cursor at the end. The submit button is located below the input field and contains the Russian word 'Отправить' (Send).

Submit button stops being disabled

Adding yet another event listener

```
// Enable button only if there is text in the input field
document.querySelector('#task').onkeyup = () => {
  document.querySelector('#submit').disabled = false;
};
```

`<input id="task"`

New Task

Отправить

1

Отправить

Disable Submit button

```
<script>
```

```
document.addEventListener('DOMContentLoaded', () => {
```

```
    // By default, submit button is disabled
```

```
    document.querySelector('#submit').disabled = true;
```

take the Submit button and disable it

New Task

Отправить

Onkeyup event

disabled



enabled

// Enable button only if there is text in the input field

```
document.querySelector('#task').onkeyup = () => {  
  document.querySelector('#submit').disabled = false;  
};
```

Submit button is still enabled

Tasks

- 13.11.2024 08:30 WEB technologies (Lecture) 153-154

New Task

Отправить



Submit button is still enabled

Отправить

Отправить

Tasks

- 13.11.2024 08:30 WEB technologies (Lecture) 153-154

New Task

Отправить



go back and
disable the Submit button again

1) We press Submit button

2) The Submit button goes back to being inactive as well.

```
// Enable button only if there is text in the input field
```

```
document.querySelector('#task').onkeyup = () =>
{
    if (document.querySelector('#task').value.length > 0)

        document.querySelector('#submit').disabled = false;

    else

        document.querySelector('#submit').disabled = true;

};
```

JavaScript has allowed us to really make our pages much more interactive

New Task

Отправить

Disabled by default

1

Отправить

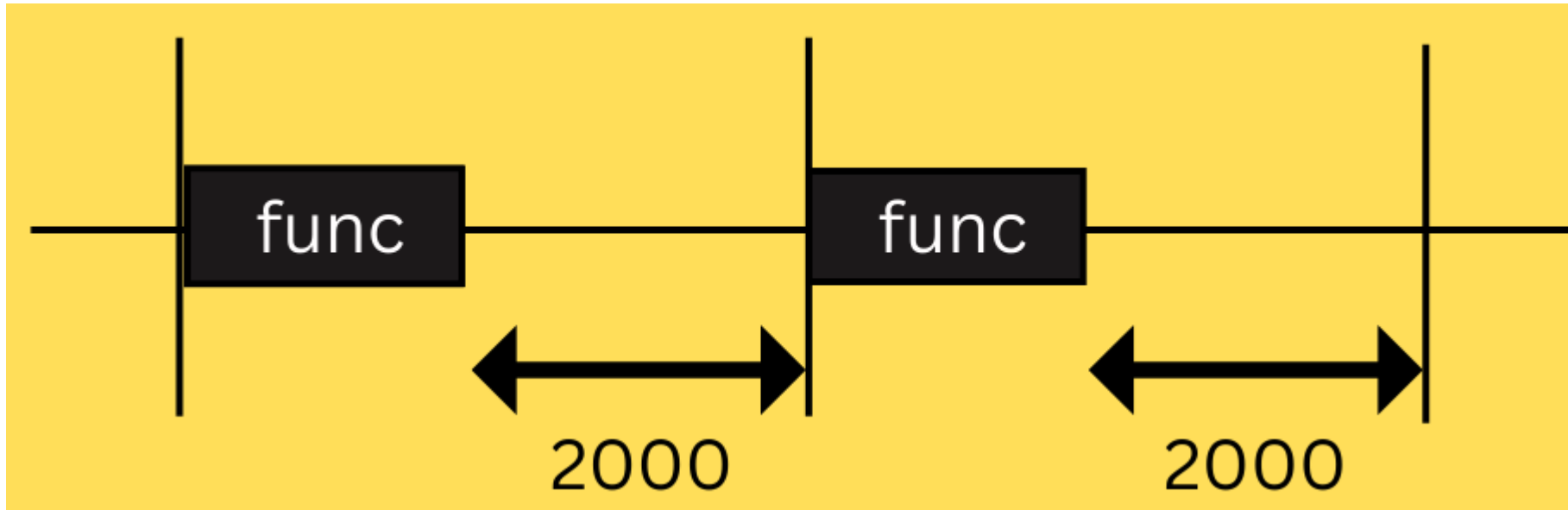
We type something in.
It's enabled.

New Task

Отправить

We delete,
and it goes back
to being disabled

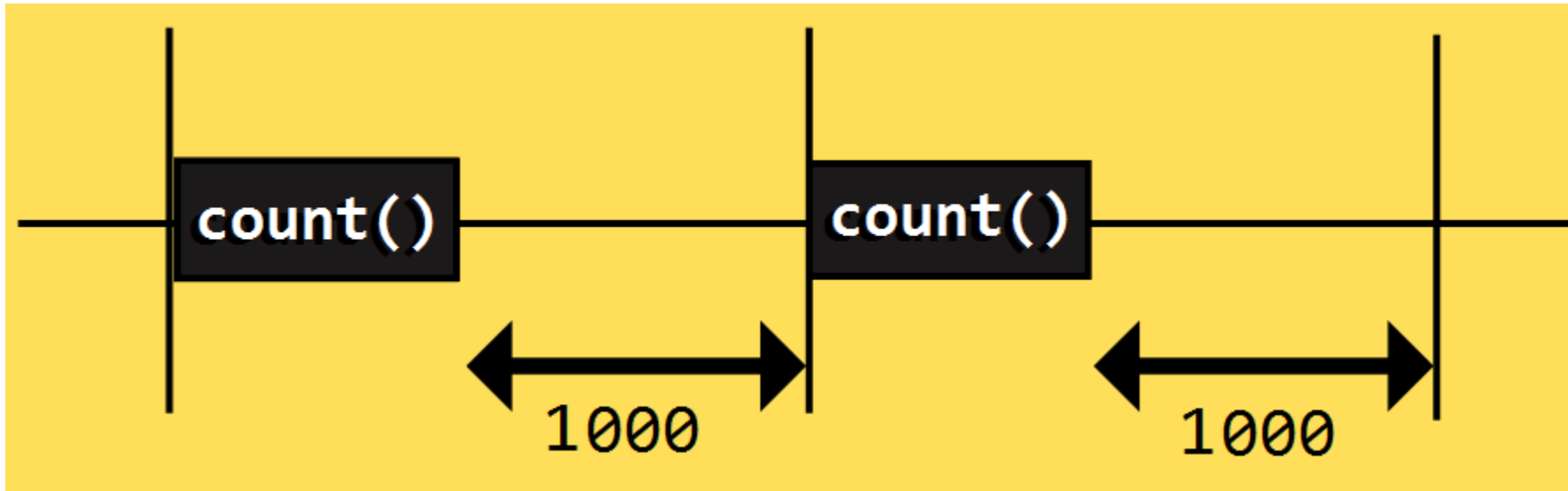
Intervals



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script>
      document.addEventListener('DOMContentLoaded', () => {
        setInterval(count, 1000);
      });

      let counter = 0;
      function count() {
        counter++;
        document.querySelector('#counter').innerHTML = counter;
      }
    </script>
    <title>Intervals</title>
  </head>
  <body>
    <h1 id="counter">0</h1>
  </body>
</html>
```

```
<h1 id="counter">0</h1>
```



```
let counter = 0;
```

JS LOCALSTORAGE

Инспектор Консоль От

Поиск в консоли

```
>> localStorage.getItem("user");
```

```
← null
```

```
>> localStorage.setItem("user", "Alex");
```

```
← undefined
```

```
>> localStorage.getItem("user");
```

```
← "Alex"
```

```
>>
```

```
<script>
```

```
let counter=0;
```

```
let counter = localStorage.getItem('counter');
```

```
<script>
```

```
// Set starting value of counter to 0
```

```
if (!localStorage.getItem('counter'))  
    localStorage.setItem('counter', 0);
```

exclamation point
just means “not”

is there already a value for counter

```
document.addEventListener('DOMContentLoaded', () => {
    document.querySelector('#counter').innerHTML =

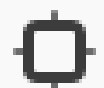
localStorage.getItem('counter');
// Count every time button is clicked
    document.querySelector('button').onclick = () => {

        // Increment current counter
        let counter = localStorage.getItem('counter');
        counter++;

        // Update counter
        document.querySelector('#counter').innerHTML = counter;
        localStorage.setItem('counter', counter);
    }
});
```



get the value of the counter.



Инспектор



Консоль



Отладчик



Поиск в консоли

```
>> localStorage.getItem("counter");
```

```
← "3"
```

```
>> localStorage.setItem("counter",0);
```

```
← undefined
```

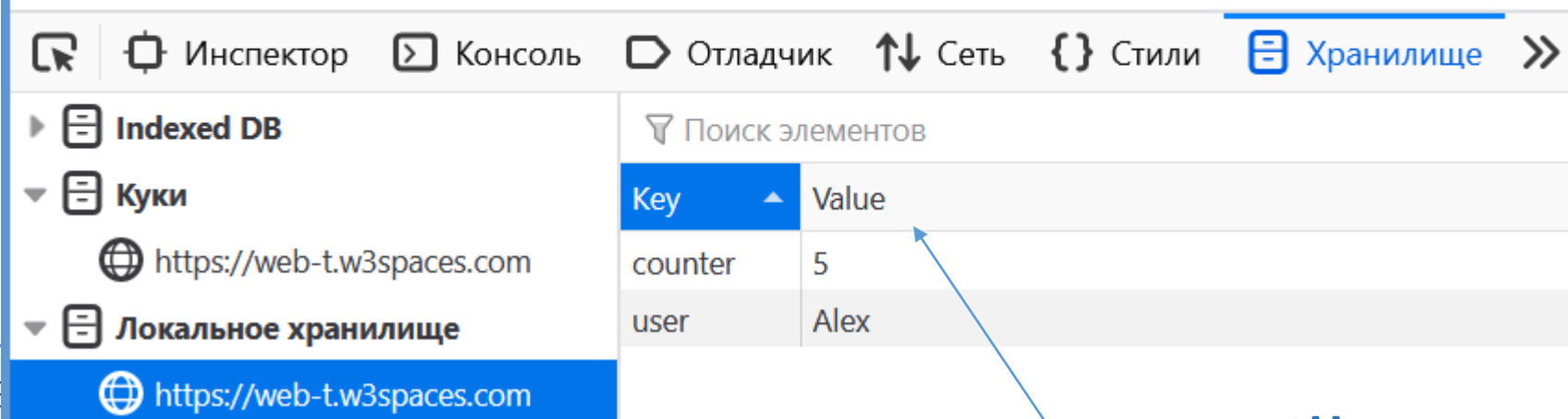
```
>> localStorage.getItem("counter");
```

```
← "0"
```

Chrome's & Mozilla's Inspectors

5

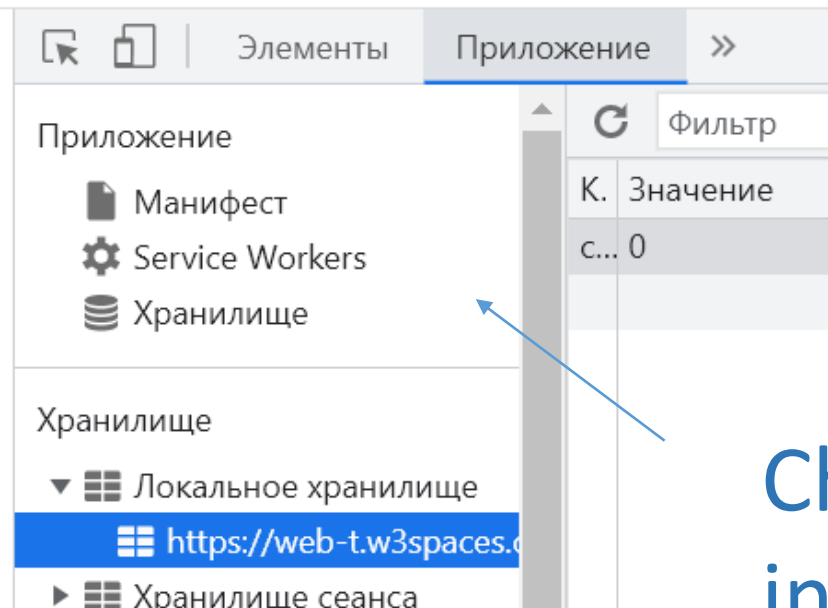
Click Here!



Mozilla
Firefox's
inspector

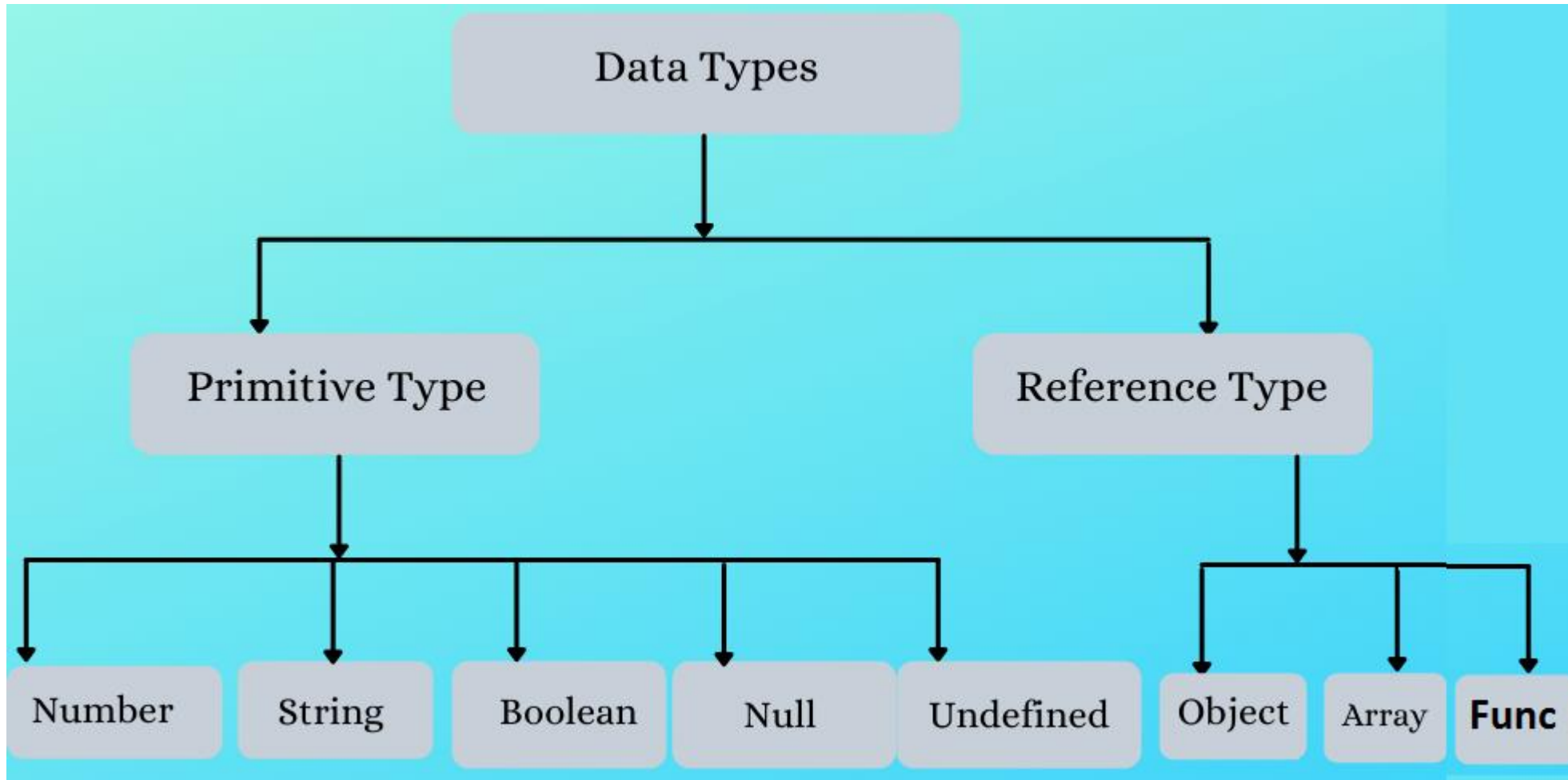
0

Click Here!



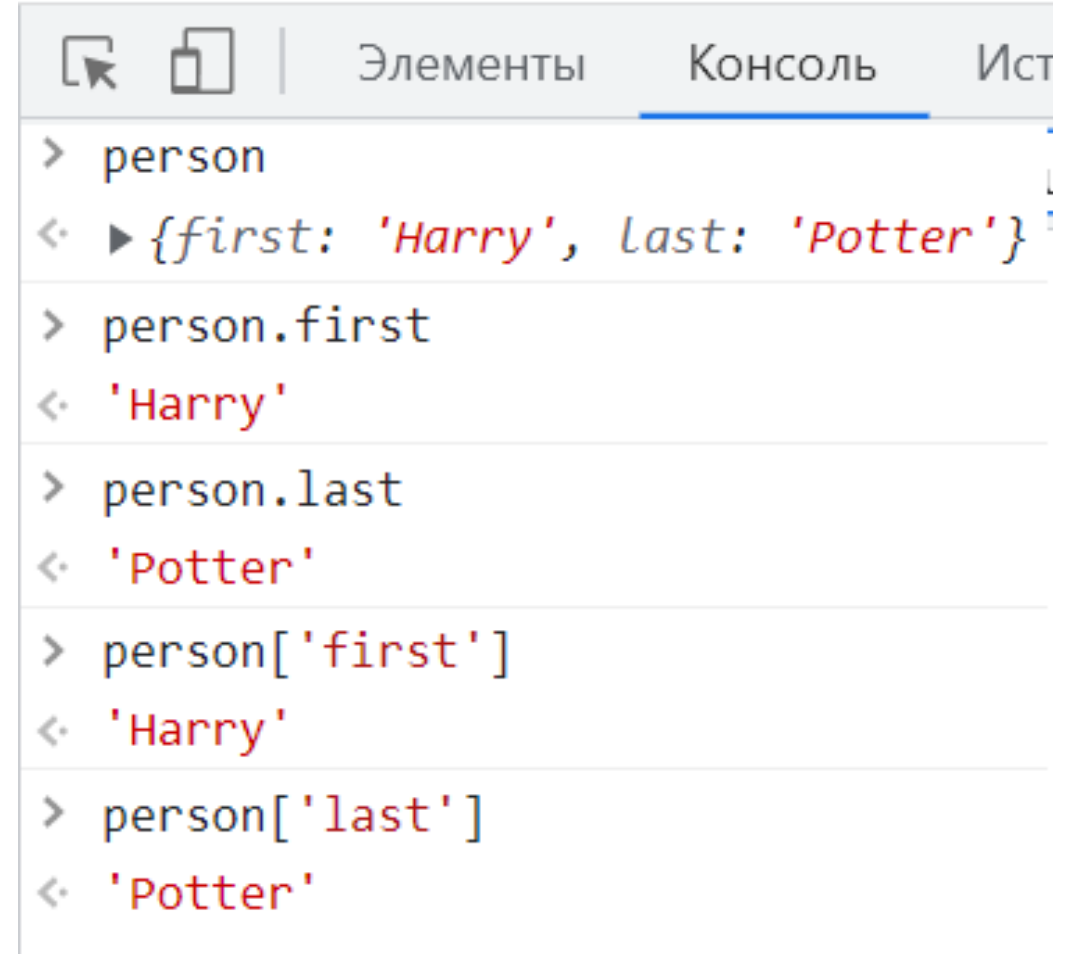
Chrome's
inspector

Variables in JavaScript

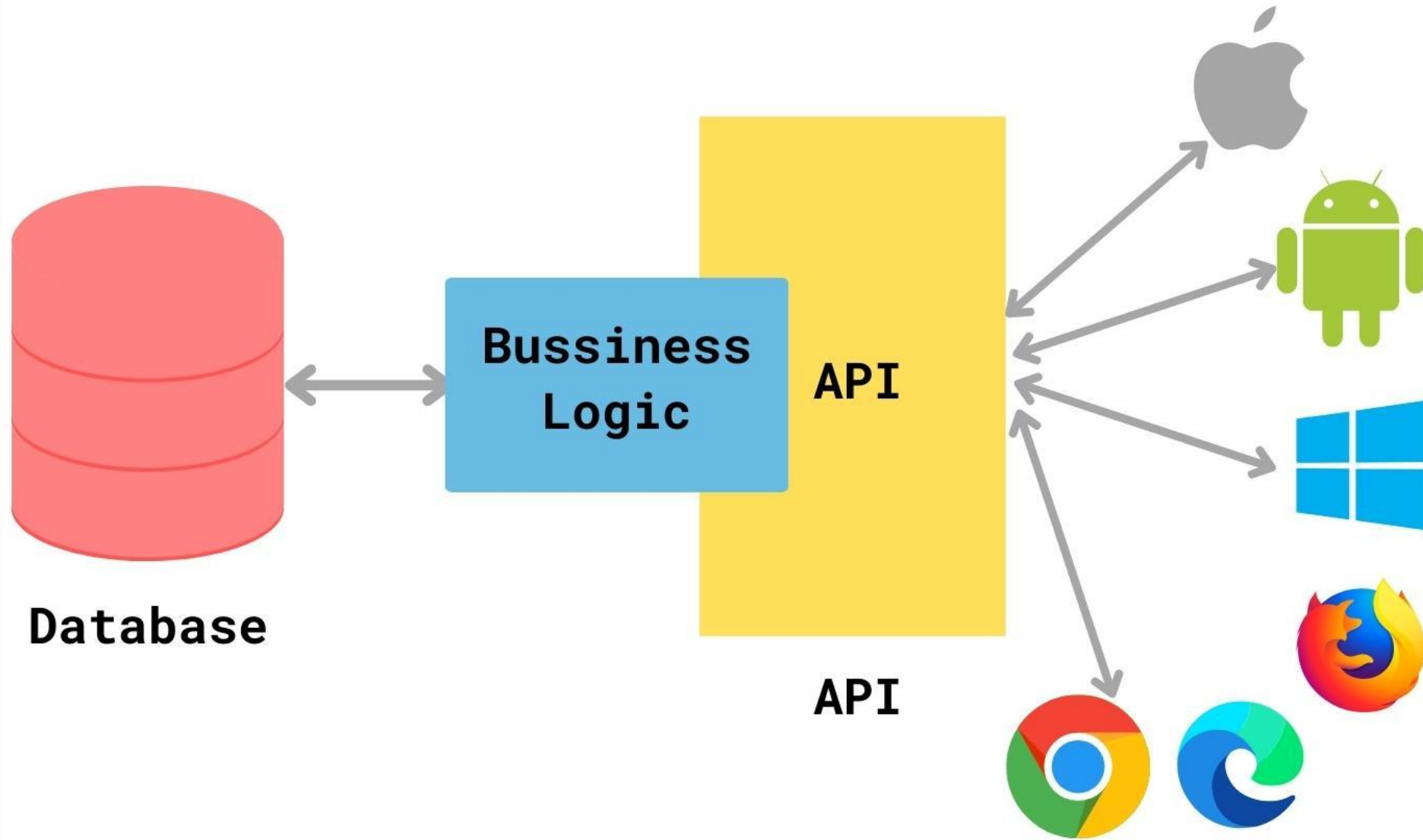


Objects

```
let person = {  
  first: 'Harry',  
  last: 'Potter'  
};
```



Application Programming Interface



JSON

```
{"managers":[  
  { "firstName":"Bill", "lastName":"Gates" },  
  { "firstName":"Mike", "lastName":"Dell" },  
]
```

<=JSON

vs

XML=>

< managers >

```
<manager>  
  <firstName>Bill</firstName>  
  <lastName>Gates</lastName>  
</manager>
```



```
<manager>  
  <firstName>Mike</firstName>  
  <lastName>Dell</lastName>  
</manager>
```



</managers>

key

colon

value

Creating a JSON string from a JavaScript object.

```
let person = {  
  first: 'Harry',  
  last: 'Potter'  
};
```

```
{  
  "first": "Harry",  
  "last": "Potter"  
}
```

```
{"managers":[  
  { "firstName":"Bill", "lastName":"Gates" },  
  { "firstName":"Mike", "lastName":"Dell" },  
]}
```

Dict

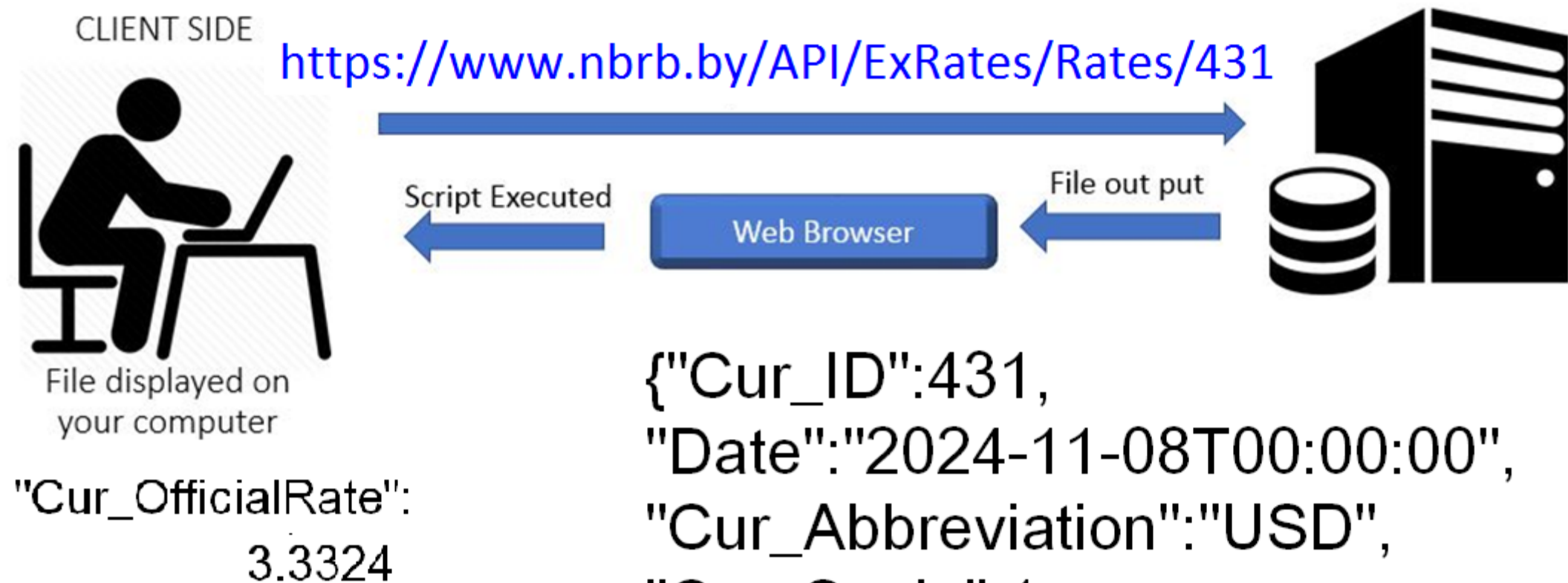


JSON

```
{'name': 'Apple'  
  'color': 'Red'  
  'quantity': 10 }
```

```
{"name": "Apple", "color":  
  "Red", "quantity": 10}
```

The process of client side scripting



\$

&

Euro

<https://www.nbrb.by/API/ExRates/Rates/431>

```
{"Cur_ID":431,  
"Date":"2024-11-08T00:00:00",  
"Cur_Abbreviation":"USD",  
"Cur_Scale":1,  
"Cur_Name":"Доллар США",  
"Cur_OfficialRate":3.3324}
```

<https://www.nbrb.by/API/ExRates/Rates/451>

```
{"Cur_ID":451,  
"Date":"2024-11-08T00:00:00",  
"Cur_Abbreviation":"EUR",  
"Cur_Scale":1,  
"Cur_Name":"Евро",  
"Cur_OfficialRate":3.6299}
```

National Bank of the Republic of Belarus

Скачать исходный код примера 

курс валюты на сегодня



Cur_ID валюты:

- {"Cur_ID":431,"Date":"2024-11-08T00:00:00","Cur_Abbreviation":"USD","Cur_Scale":1,"Cur_Name":"Доллар США","Cur_OfficialRate":3.3324}

<https://www.nbrb.by/apihelp/exrates/sample>

<https://www.nbrb.by/API/ExRates/Rates/431>

<https://web-t.w3spaces.com/currency.html>



`<button onclick='USD()'>$</button>`



`<button onclick='Euro()'>€</button>`

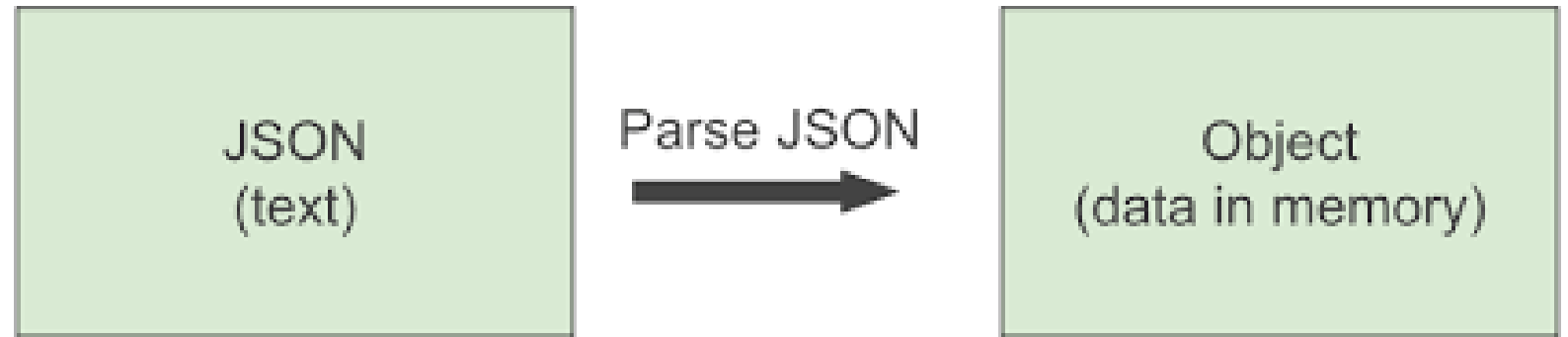
```
function USD()
{
var u=
"https://www.nbrb.by/API/ExRates/Rates/431";
$.getJSON(u)
.done(function (data) {
$('#price').html(data.Cur_OfficialRate);
alert(data.Cur_OfficialRate);
});
}
```

```
function Euro()
{
var u=
"https://www.nbrb.by/API/ExRates/Rates/451";
$.getJSON(u)
.done(function (data) {
$('#price').html(data.Cur_OfficialRate);
alert(data.Cur_OfficialRate);
});
}
```

`<h1 id="price"></h1>`

```
<script src=
"https://ajax.googleapis.com/ajax/libs/jquery/3
.4.1/jquery.min.js">
</script>
```

```
myObj = JSON.parse(minskJSON)
```



```
> minskJSON
```

```
< '{"name":"Ira", "age":21, "city":"Minsk"}'
```

```
> myObj
```

```
< {name: 'Ira', age: 21, city: 'Minsk'}
```

```
JSON.parse(minskJSON);
```

```
<p id="demo"></p>
```

```
<script>
```

```
var minskJSON = '{"name":"Ira", "age":21, "city":"Minsk"}';
```

```
var myObj = JSON.parse(minskJSON);
```

```
document.getElementById("demo").innerHTML =
```

```
"<b>" + myObj.name +
```

```
"</b> from <b>" +
```

```
myObj.city + "</b>";
```

```
</script>
```

Stringify of an object into a JSON



JSON.stringify() - converts object to a JavaScript string.

["John","Peter","Sally","Jane"]

JSON
(text)

←
Stringify JSON

["John", "Peter", "Sally", "Jane"]

Object
(data in memory)

<body>

<p id="demo"></p>

<script>

var arr = ["John", "Peter", "Sally", "Jane"];

var myJSON = JSON.stringify(arr); //["John","Peter","Sally","Jane"]

document.getElementById("demo").innerHTML = myJSON;

</script>

</body>

```
<html>
<body>

<p id="demo"></p>
```

```
<script>
```

```
var obj = { name: "JSON Lcture",
            now: (new Date()).getFullYear()+1900};
```

```
var mJSON = JSON.stringify(obj);
```

```
document.getElementById("demo").innerHTML = mJSON;
```

```
</script>
```

```
</body>
```

```
</html>
```

```
> obj
```

```
< ▶ {name: 'JSON Lecture', now: 2024}
```

```
> mJSON
```

```
< '{"name":"JSON Lecture","now":2024}'
```