Кросс-платформенная разработка

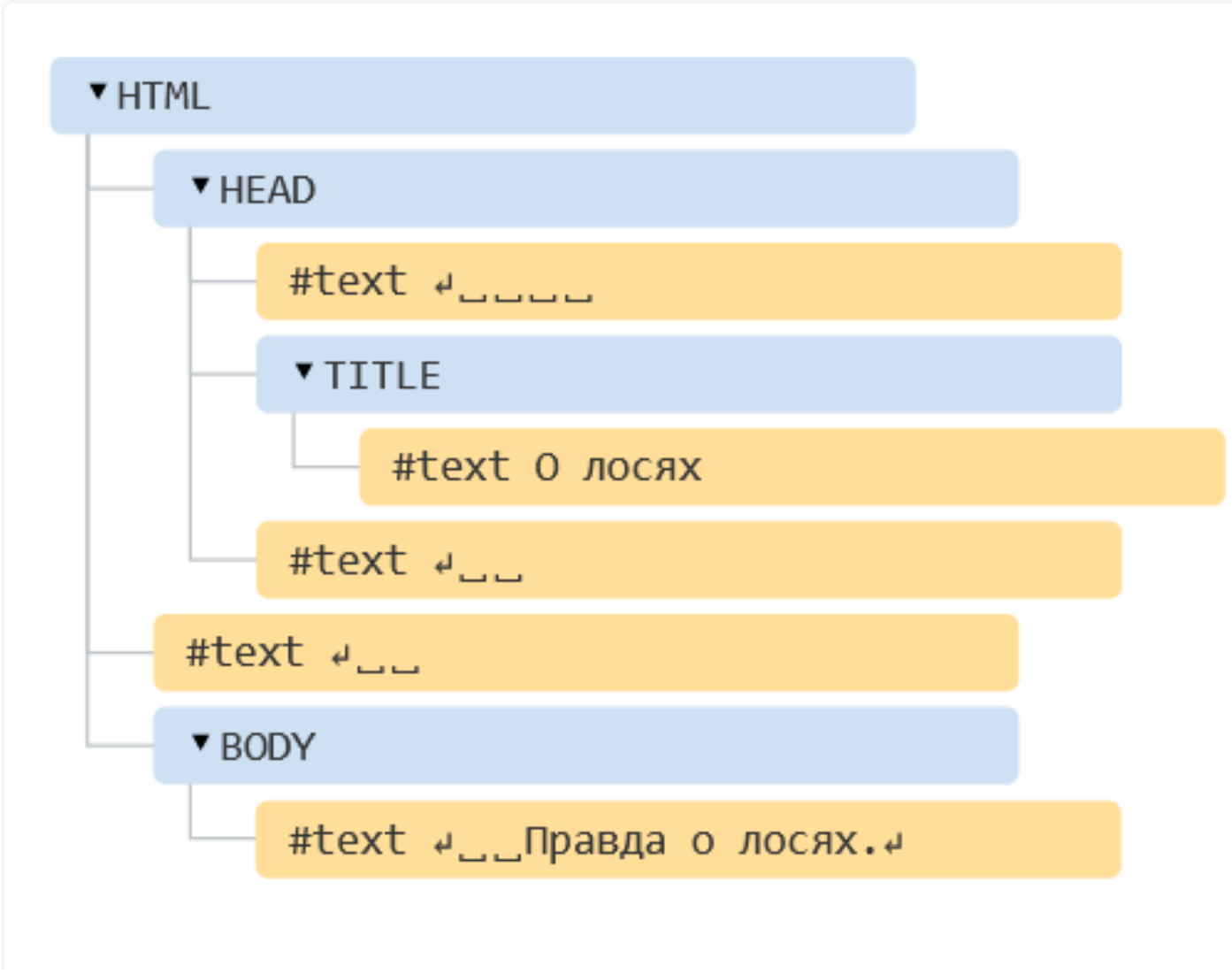
Lecture 7

Topics

- DOM
- Теги

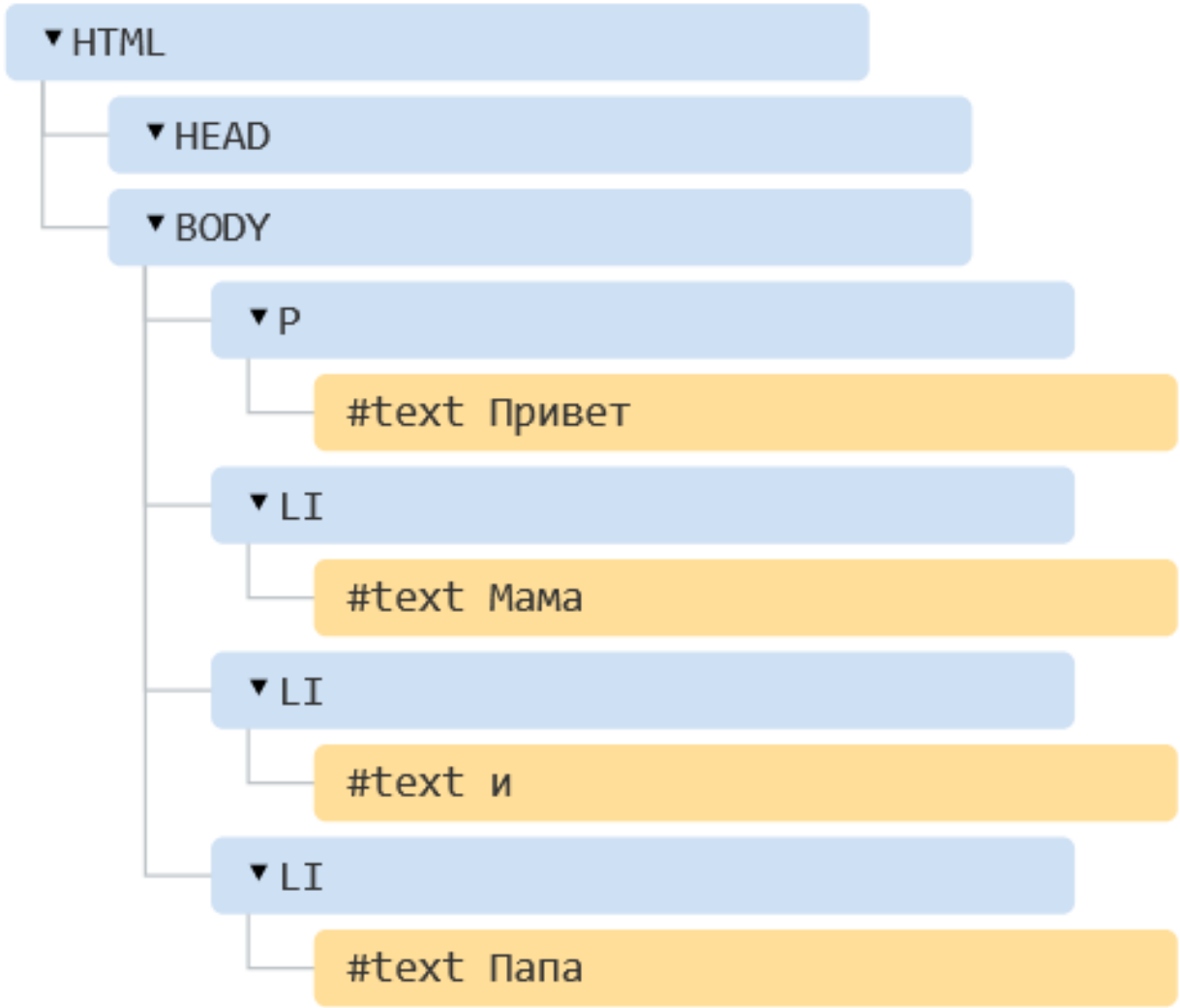
HTML => DOM

```
<!DOCTYPE HTML>
<html>
<head>
   <title>О лосях</title>
</head>
<body>
Правда о лосях.
</body>
</html>
```



Автоисправление

```
<p>Привет
    <li>Мама
    <li>и
<li>Папа
```



Доступ из JS

```
<button id="demoButton">
    Создать запись
</button>
```

```
document.getElementById('demoButton').onclick = addSomething;

demoButton.onclick = addSomething;
```

Глобальные имена

```
let demoButton = 5;

alert(demoButton); // 5
```

Cheat Sheet

Метод	Ищет по...	Ищет внутри элемента?	Возвращает живую коллекцию?
`querySelector`	CSS-selector	✓	-
`querySelectorAll`	CSS-selector	✓	-
`getElementById`	`id`	-	-
`getElementsByName`	`name`	-	✓
`getElementsByTagName`	tag or `'*'`	✓	✓
`getElementsByClassName`	class	✓	✓

Атрибуты и свойства

```
<body id="test" something="non-standard">
<script>
    alert(document.body.id); // test
    alert(document.body.something); // undefined
</script>
</body>
```

Атрибуты могут отличаться

```
<body id="body" type="...">
<input id="input" type="text">
  <script>
    alert(input.type); // text
    alert(body.type); // undefined
  </script>
</body>
```

Доступ к нестандартным атрибутам

```
<body something="non-standard">
<script>
  alert(document.body.getAttribute('something'));
</script>
</body>
```

Работа с атрибутами

```
<div id="elem" about="Elephant"></div>

<script>
  alert( elem.getAttribute('About') ); // (1) 'Elephant', чтение

  elem.setAttribute('Test', 123); // (2), запись

  alert( elem.outerHTML ); // (3), посмотрим, есть ли атрибут в HTML (да)

  for (let attr of elem.attributes) { // (4) весь список
    alert( `${attr.name} = ${attr.value}` );
  }
</script>
```

Синхронизация...

```
<script>
    let input = document.querySelector('input');

    // атрибут => свойство
    input.setAttribute('id', 'id');
    alert(input.id); // id (обновлено)

    // свойство => атрибут
    input.id = 'newId';
    alert(input.getAttribute('id')); // newId (обновлено)
</script>
```

… работает не всегда

```
<script>
    let input = document.querySelector('input');

    // атрибут => значение
    input.setAttribute('value', 'text');
    alert(input.value); // text

    // свойство => атрибут
    input.value = 'newValue';
    alert(input.getAttribute('value')); // text (не обновилось!)
</script>
```

Типы свойств

```
<input id="input" type="checkbox" checked> checkbox

<script>
  alert(input.getAttribute('checked')); // значение атрибута: пустая строка
  alert(input.checked); // значение свойства: true
</script>
```

Свойства как объекты

```
<div id="div" style="color:red;font-size:120%">Hello</div>

<script>
   // строка
   alert(div.getAttribute('style')); // color:red;font-size:120%

   // объект
   alert(div.style); // [object CSSStyleDeclaration]
   alert(div.style.color); // red
</script>
```

Преобразование свойств

```
<a id="a" href="#hello">link</a>
<script>
  // атрибут
  alert(a.getAttribute('href')); // #hello

  // свойство
  alert(a.href ); // полный URL в виде http://site.com/page#hello
</script>
```

Кастомные атрибуты как инструмент

```
<!-- пометить div, чтобы показать здесь поле "name" -->
<div show-info="name"></div>
<!-- а здесь возраст "age" -->
<div show-info="age"></div>
```

Обращение из JS

```
<script>
   // код находит элемент с пометкой и показывает запрошенную информацию
   let user = {
      name: "Pete",
      age: 25
   };

   for(let div of document.querySelectorAll('[show-info]')) {
      // вставить соответствующую информацию в поле
      let field = div.getAttribute('show-info');
      div.innerHTML = user[field]; // сначала Pete в name, потом 25 в age
   }
</script>
```

Стили для кастомных атрибутов

```
<style>
   /* стили зависят от пользовательского атрибута "order-state" */
   .order[order-state="new"] {
      color: green;
   }

   .order[order-state="pending"] {
      color: blue;
   }

   .order[order-state="canceled"] {
      color: red;
   }
</style>
```

Применение стилей

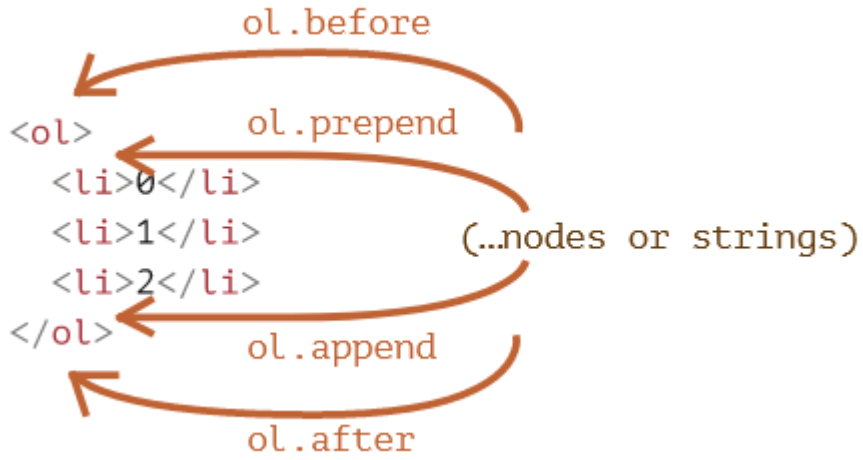
```
<div class="order" order-state="new">
    A new order.
</div>

<div class="order" order-state="pending">
    A pending order.
</div>

<div class="order" order-state="canceled">
    A canceled order.
</div>
```

Dataset

```
<body data-about="Elephants">
<script>
    alert(document.body.dataset.about); // Elephants
</script>
```

Динамические изменения

```
<script>
   ol.before('before'); // вставить строку "before" перед <ol>
   ol.after('after'); // вставить строку "after" после <ol>

   let liFirst = document.createElement('li');
   liFirst.innerHTML = 'prepend';
   ol.prepend(liFirst); // вставить liFirst в начало <ol>

   let liLast = document.createElement('li');
   liLast.innerHTML = 'append';
   ol.append(liLast); // вставить liLast в конец <ol>
</script>
```



Фрагменты

```
function getListContent() {
    let fragment = new DocumentFragment();

    for(let i=1; i<=3; i++) {
        let li = document.createElement('li');
        li.append(i);
        fragment.append(li);
    }

    return fragment;
}

ul.append(getListContent()); // (*)
```

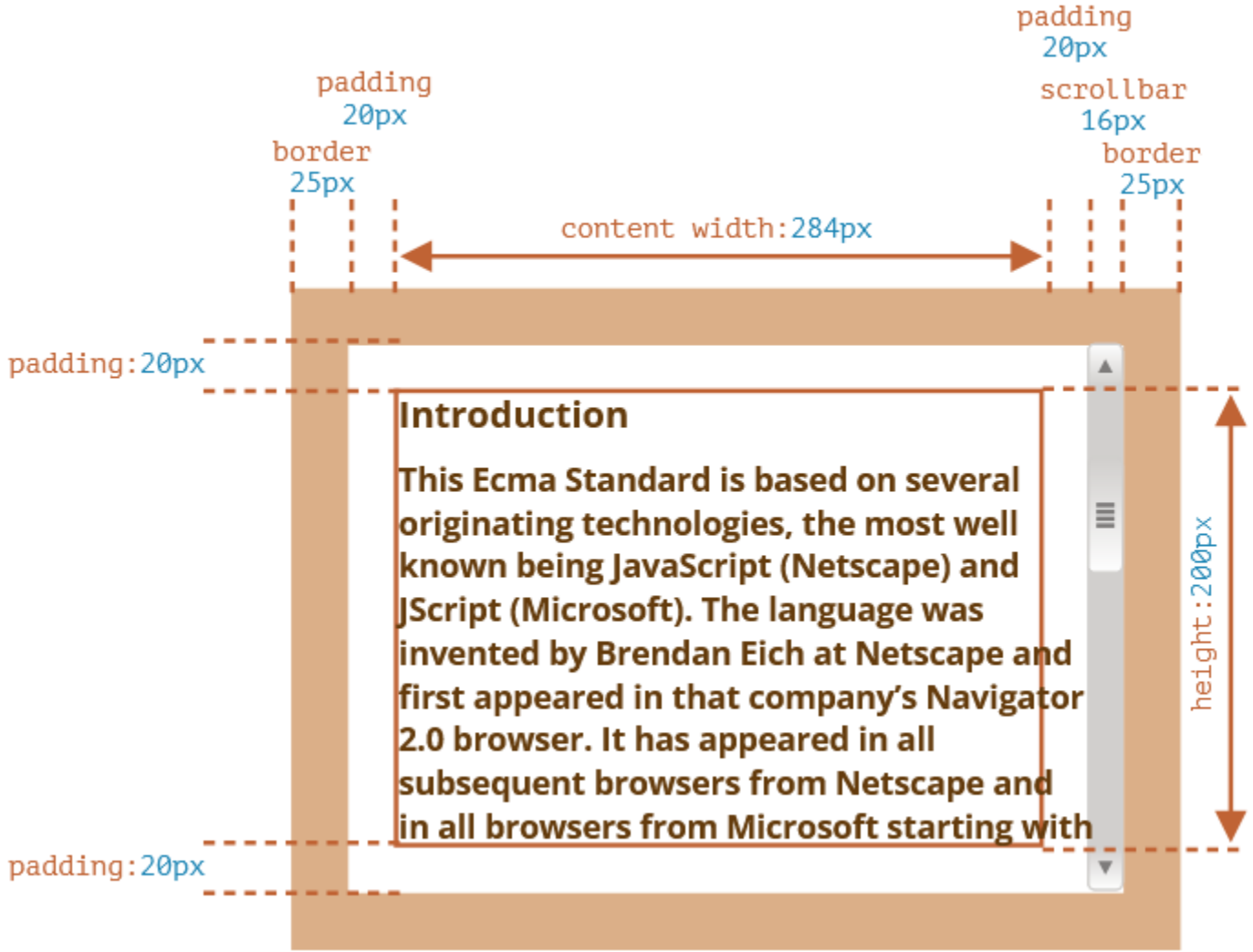
```
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
</ul>
```

Без фрагментов

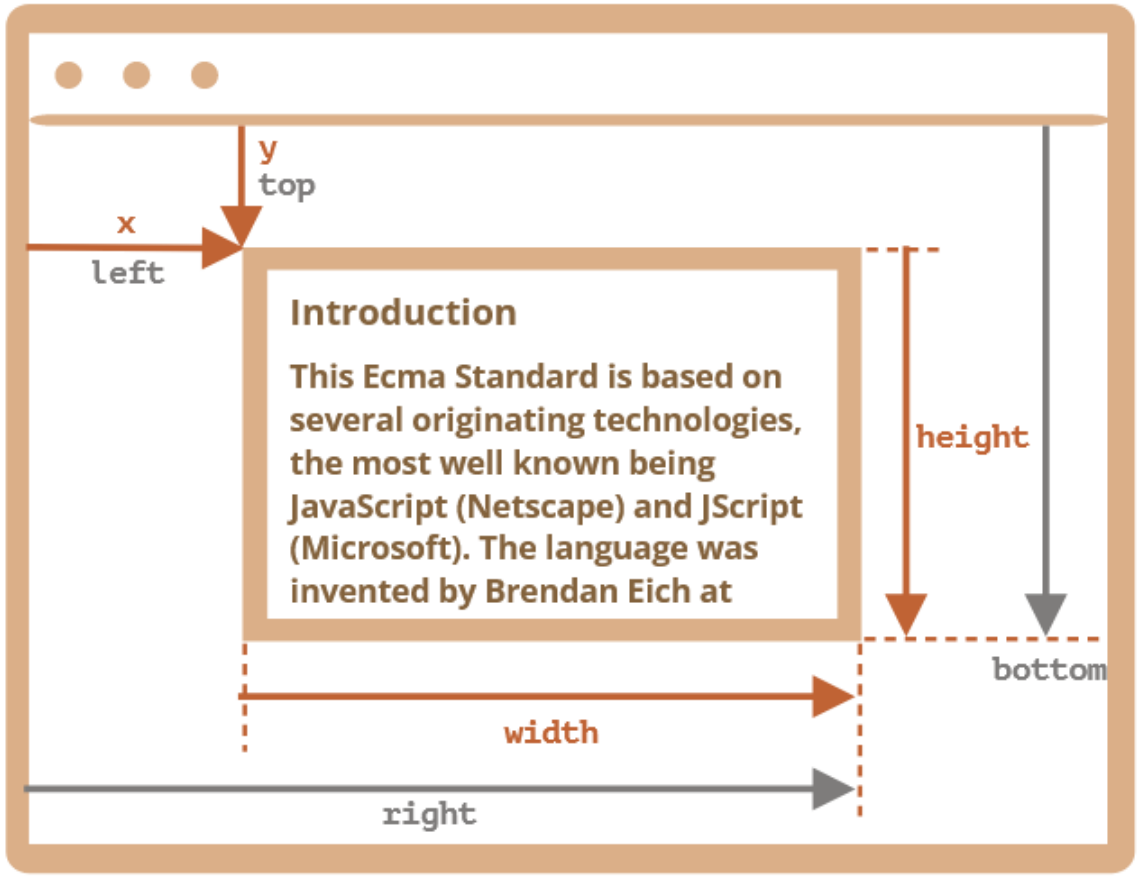
```
function getListContent() {
    let result = [];

    for(let i=1; i<=3; i++) {
        let li = document.createElement('li');
        li.append(i);
        result.push(li);
    }

    return result;
}

ul.append(...getListContent()); // append + оператор "..." = друзья!
```

Размеры элементов

```
<div id="example">
    ...Текст...
</div>
<style>
    #example {
        width: 300px;
        height: 200px;
        border: 25px solid #E8C48F;
        padding: 20px;
        overflow: auto;
    }
</style>
```



Координаты


y
top
x
left
Introduction
This Ecma Standard is based on several originating technologies, the most well known being JavaScript (Netscape) and JScript (Microsoft). The language was invented by Brendan Eich at
height
bottom
width
right

Единицы измерения - Абсолютные

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

Единицы измерения - Относительные

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

https://html5book.ru/

- Где-то надо учиться HTML/CSS

UI библиотеки

- **Bootstrap**
- Bulma
- Foundation
- UIKit
- Chakra
- **Tailwind**

Проектируем индивидуальное задание

References

- https://javascript.info/
- https://developer.mozilla.org/
- https://www.w3schools.com