

Факультатив ФКН

РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА БАЗЕ ПЛАТФОРМЫ



Преподаватель

Папулин Сергей Юрьевич (papulin_hse@mail.ru)

Ассистент

Цырлин Никита (nstsyrlin@gmail.com)



Занятие 4. Разработка веб-приложений. JavaScript, jQuery, TAU



Основные темы

➤ JavaScript и jQuery

➤ Шаблоны



JavaScript и jQuery



JavaScript



JS

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

```
<script>
```

```
    document.body.innerHTML = "Hello JavaScript!";
```

```
</script>
```



JS

```
var length = 16;           // Number
var lastName = "Johnson"; // String
var cars = ["Saab", "Volvo", "BMW"]; // Array
var x = {firstName:"John", lastName:"Doe"}; // Object
```



JS

```
var x;                // Now x is undefined
var x = 5;            // Now x is a Number
var x = "John";       // Now x is a String
```

```
var x = 16 + 4 + "Volvo";
```

```
var x = "Volvo" + 16 + 4;
```



JavaScript. Переменные и типы данных

JS

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement



JavaScript. Переменные и типы данных

JS

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
<<=	x <<= y	x = x << y
>>=	x >>= y	x = x >> y
>>>=	x >>>= y	x = x >>> y
&=	x &= y	x = x & y
^=	x ^= y	x = x ^ y
=	x = y	x = x y



JavaScript. Массивы

JS

```
var array-name = [item1, item2, ...];
```

```
var array-name = new Array(item1, item2, ...);
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits.push("Lemon"); // adds a new element (Lemon) to fruits
```

```
fruits[fruits.length] = "Lemon"; // adds a new element (Lemon) to fruits
```

```
arr.splice(index[, deleteCount, elem1, ..., elemN])
```



JavaScript. Функция

JS

```
function name(parameter1, parameter2, parameter3) {  
    code to be executed  
}
```

```
var x = function (parameter1, parameter2, parameter3) {code to be executed};
```



JavaScript. Функция

JS

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}  
document.body.innerHTML = toCelsius(77);
```

```
var x = function (a, b) {return a * b};  
var z = x(4, 3);
```

```
(function () {  
    var x = "Hello!!";    // I will invoke myself  
})();
```



JavaScript. Объекты



JS

```
var objectName = {  
  property1: value,  
  property2: value,  
  property3: value,  
  method1: function() { code lines },  
  method2: function() { code lines }  
};
```

```
objectName[property1] = value
```

```
objectName.property1 = value
```

```
objectName.newProperty = value
```

```
delete objectName.property
```



JavaScript. Объекты

JS

```
var person = {  
  firstName:"John",  
  lastName:"Doe",  
  age:50,  
  eyeColor:"blue"  
};
```

```
var person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```



JavaScript. Объекты. Конструктор

JS

```
function person(first, last, age, eye) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
    this.eyeColor = eye;  
}  
  
var myFather = new person("John", "Doe", 50, "blue");
```



JavaScript. Объекты. Прототип

JS

```
function Person(first, last, age, eyecolor) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
    this.eyeColor = eyecolor;  
}
```

```
var myFather = new Person("John", "Doe", 50, "blue");  
  
myFather.nationality = "English";
```

```
Person.prototype.nationality = "English";
```

```
Person.prototype.name = function() {  
    return this.firstName + " " + this.lastName;  
};
```




JavaScript. Условный оператор



JS

```
variablename = (condition) ? value1:value2
```

```
if (condition1) {  
    block of code to be executed if condition1 is true  
} else if (condition2) {  
    block of code to be executed if the condition1 is false and  
condition2 is true  
} else {  
    block of code to be executed if the condition1 is false and  
condition2 is false  
}
```

http://www.w3schools.com/js/js_if_else.asp

http://www.w3schools.com/js/js_comparisons.asp



JavaScript. Условный оператор

JS

```
var voteable = (age < 18) ? "Too young":"Old enough";
```

```
age = Number(age);  
if (isNaN(age)) {  
    voteable = "Error in input";  
} else {  
    voteable = (age < 18) ? "Too young" : "Old enough";  
}
```



JavaScript. Условный оператор

JS

Operator	Description	Comparing	Returns
==	equal to	x == 8	false
		x == 5	true
		x == "5"	true
===	equal value and equal type	x === 5	true
		x === "5"	false
!=	not equal	x != 8	true
!==	not equal value or not equal type	x !== 5	false
		x !== "5"	true
		x !== 8	true
>	greater than	x > 8	false
<	less than	x < 8	true
>=	greater than or equal to	x >= 8	false
<=	less than or equal to	x <= 8	true



JavaScript. Switch оператор

JS

```
switch(expression) {  
    case n:  
        code block  
        break;  
    case n:  
        code block  
        break;  
    default:  
        default code block  
}
```



JavaScript. Условный оператор

JS

Operator	Description
&&	logical and
	logical or
!	logical not



JavaScript. Цикл for



JS

```
for (initial value; condition; change) {  
    code block to be executed  
}
```

```
for (item in array) {  
    code block to be executed  
}
```



JavaScript. Цикл for

JS

```
for (i = 0; i < 5; i++) {  
    text += "The number is " + i + "<br>";  
}
```

```
var person = {fname:"John", lname:"Doe", age:25};
```

```
var text = "";  
var x;  
for (x in person) {  
    text += person[x];  
}
```



JavaScript. Цикл while



JS

```
while (condition) {  
    code block to be executed  
}
```

```
do {  
    code block to be executed  
}  
while (condition);
```




JavaScript. Цикл while

JS

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

```
do {  
    text += "The number is " + i;  
    i++;  
}  
while (i < 10);
```



JavaScript. Break/Continue

JS

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { continue; } //break the same  
    text += "The number is " + i + "<br>";  
}
```

label:
statements

`break labelname;`

`continue labelname;`

```
var cars = ["BMW", "Volvo", "Saab", "Ford"];  
list: {  
    text += cars[0] + "<br>";  
    text += cars[1] + "<br>";  
    text += cars[2] + "<br>";  
    break list;  
    text += cars[3] + "<br>";  
    text += cars[4] + "<br>";  
    text += cars[5] + "<br>";  
}
```



Выбор элементов



document.**getElementById**("id")

\$("#**id**")

CSS стиль

document.**getElementsByTagName**("tag")

\$("**tag**")

document.**getElementsByClassName**("class")

\$("**.class**")

document.**querySelector**("**.class**")

CSS стиль

#id

tag

Источники

[Javascript](#)

[jQuery](#)

document.**querySelectorAll**("**.class**", "tag", ...)



События



```
element.addEventListener(event, function, useCapture)
```

```
element.removeEventListener(event, function)
```

```
element.ontevent = function
```



```
$("element").event(function)
```

```
$("element").on(event, selector, data, function)
```

```
$("element").off(function)
```

```
$("element").on({  
    event1: function  
    },  
    event2: function  
    })
```

Примеры
[Multiple functions](#)
[Toggle](#)

Ссылки
[Javascript](#)
[jQuery](#)



JavaScript vs jQuery

[C4.2 Page Load Event](#)

[C4.3 Element Group Event](#)

[C4.4 Multiple Function Element Event](#)

[C4.2 Get Set Text](#)

[C4.4 Input Value Text](#)

[C4.3 InnerHTML](#)

[C4.4 Get Set Attribute](#)

[C4.5 Style Attribute](#)

[C4.3 Add Class](#)

[C4.5 Append Element](#)

[C4.4 Insert Before After](#)

[C4.3 Prepend/Append vs Before/After](#)

[C4.5 Wrap](#)

[C4.6 Replace](#)

[C4.3 JS vs JQ Toggle](#)

[C4.6 Show Hide Content](#)

[C4.6 Bootstrap jQuery UI](#)

[C4.6 Bootstrap jQuery Mobile](#)



Button API

Button

[Ссылка на API](#)

`<button id="button">Button</button>`

`<div id="button" class="ui-btn">Button</div>`

Свойство	Тип	Значение по умолчанию	Описание
data-icon	string null	null	Если свойство используется, то необходимо добавить class "ui-icon-" в элемент и прописать его в css.
data-iconpos	"left" "right" "top" "bottom" "notext" null	"left"	Положение изображения
data-inline	boolean	false	Если true, то display = "inline" CSS свойство.
data-disabled	boolean	false	Если true, то кнопка неактивна
data-disabled	"circle" "nobg" null	null	Стиль кнопки

Метод	Описание
Button disable ()	Перевод кнопки в неактивное состояние
Button enable ()	Перевод кнопки в активное состояние
Button refresh ()	Обновить отображение кнопки
string value (string? value)	Получить/установить значение



Checkbox and ToogleSwitch API



[Ссылка на API](#)

```
<input type="checkbox" name="checkbox-1" id="checkbox-1"/>  
<label for="checkbox-1">Apple</label>
```

```
<input type="checkbox" name="toggle-1" id="toggle-1"/>  
<label for="toggle-1" class="ui-toggleswitch">Apple</label>
```

Метод	Описание
Checkbox/ToggleSwitch disable ()	Перевод чекбокса/переключателя в неактивное состояние
Checkbox/ToggleSwitch enable ()	Перевод чекбокса/переключателя в активное состояние
string value (string? value)	Получить/установить значение

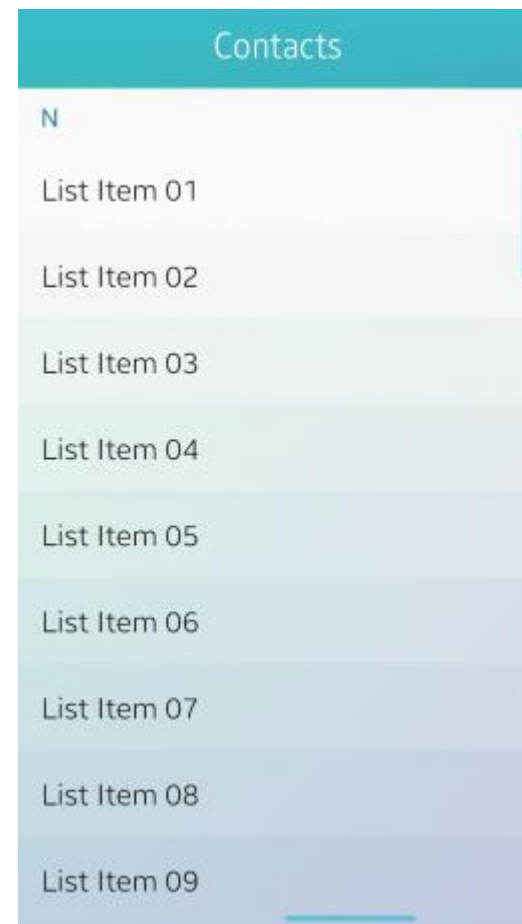


Listview API

```
<ul class="ui-listview">  
  <li class="ui-li-static">list item1</li>  
  <li class="ui-li-static">list item2</li>  
  <li class="ui-li-static">list item3</li>  
  <li class="ui-li-static">list item4</li>  
</ul>
```

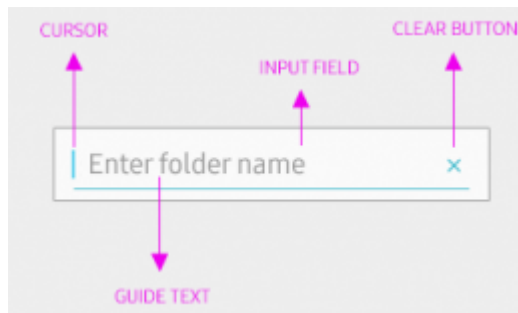
[Ссылка на API](#)

```
<ul class="ui-listview">  
  <li class="ui-li-anchor"><a href="page1.html">Page1</a></li>  
  <li class="ui-li-anchor"><a href="page2.html">Page2</a></li>  
  <li class="ui-li-anchor"><a href="page3.html">Page3</a></li>  
  <li class="ui-li-anchor"><a href="page4.html">Page4</a></li>  
</ul>
```





TextInput API



[Ссылка на API](#)

```
<input type="text/number/password/email/tel/url" name="texttype" id="texttype"/>
```

```
<textarea type="text" name="textarea" id="textarea"></textarea>
```

Свойство	Тип	Значение по умолчанию	Описание
data-clear-btn	boolean	false	Показывать кнопку очистки поля
data-text-line	boolean	true	Подчеркивать текст

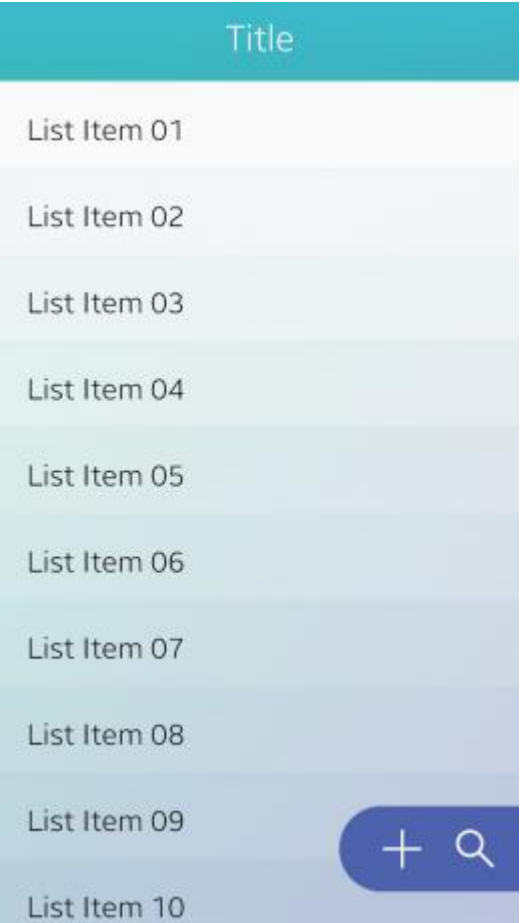
Метод	Описание
TextInput disable ()	Перевод поля в неактивное состояние
TextInput enable ()	Перевод поля в активное состояние



FloatingAction API

```
<div class="ui-floatingactions" id="floating">  
  <button class="ui-floatingactions-item" data-icon="floating-add"/>  
  <button class="ui-floatingactions-item" data-icon="floating-search"/>  
</div>
```

[Ссылка на API](#)



Свойство	Тип	Значение по умолчанию	Описание
data-from-rgba	string	"rgba(66, 162, 207, 1)"	Цвет, если расположена слева экрана
data-to-rgba	string	"rgba(54, 132, 168, 1)"	Цвет, если расположена справа экрана
data-opacity	number	0.9	Прозрачность при клике
data-duration	number	300	Длительность анимации в миллисекундах





Wearable Bottom Button API



```
<footer class="ui-footer ui-bottom-button">  
  <button id="bottomBtn" class="ui-btn">Search</button>  
</footer>
```

[Ссылка на API](#)

Свойство	Тип	Значение по умолчанию	Описание
data-icon	string null	null	Если свойство используется, то необходимо добавить class "ui-icon-" в элемент и прописать его в css.
data-iconpos	"left" "right" "top" "bottom" "notext" null	"left"	Положение изображения
data-inline	boolean	false	Если true, то display = "inline" CSS свойство.
data-disabled	boolean	false	Если true, то кнопка неактивна
data-disabled	"circle" "nobg" null	null	Стиль кнопки

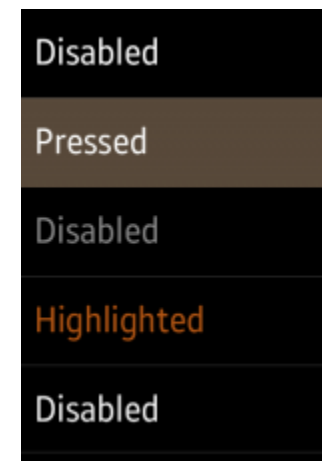
Метод	Описание
Button disable ()	Перевод кнопки в неактивное состояние
Button enable ()	Перевод кнопки в активное состояние
Button refresh ()	Обновить отображение кнопки
string value (string? value)	Получить/установить значение



Wearable ListView API

[Ссылка на API](#)

```
<ul class="ui-listview">  
  <li><a href="#">1line</a></li>  
  <li> <a href="#">2line</a></li>  
  <li><a href="#">3line</a></li>  
  <li><a href="#">4line</a></li>  
  <li><a href="#">5line</a></li>  
</ul>
```





https://developer.tizen.org/dev-guide/2.4.0/org.tizen.web.apireference/html/ui_fw_api/Mobile_UIComponents/mobile_component_list.htm

https://developer.tizen.org/dev-guide/2.3.2/org.tizen.web.apireference/html/ui_fw_api/wearable/widgets/widget_reference.htm



Источники

<http://www.w3schools.com/>

<http://htmlbook.ru/>

<https://www.html5rocks.com/en/tutorials/internals/howbrowserswork>

<https://www.smashingmagazine.com/wp-content/uploads/images/css3-cheat-sheet/css3-cheat-sheet.pdf>



СПАСИБО ЗА ВНИМАНИЕ