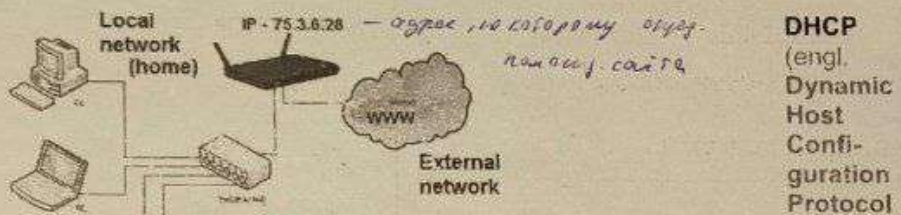
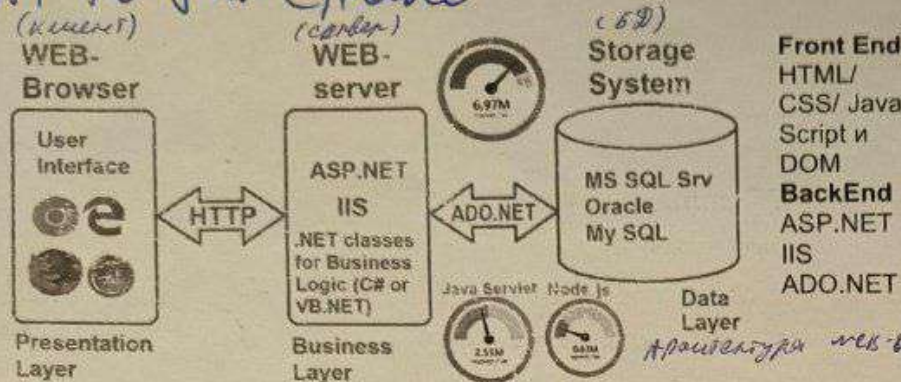


40,000 8.9.25
+0.1 to fin Grade



tracert - tracerout (in Apple) www.geotracerroute.com

172 . 16 . 254 . 1
↓ ↓ ↓ ↓
10101100.00010000.11111110.00000001

4 billion addresses
- 4.294.967.296

DNS - clearer address.
Domain Name System

217.21.43.40 - sbmt.by, sbmt.bsu.by sb.bsu.by
ping sb.bsu.by tracert sb.bsu.by <https://geotracerroute.com/>

+375 (29) 254 07 92 - ANDREY O. YAROSHEVICH
ID Domain

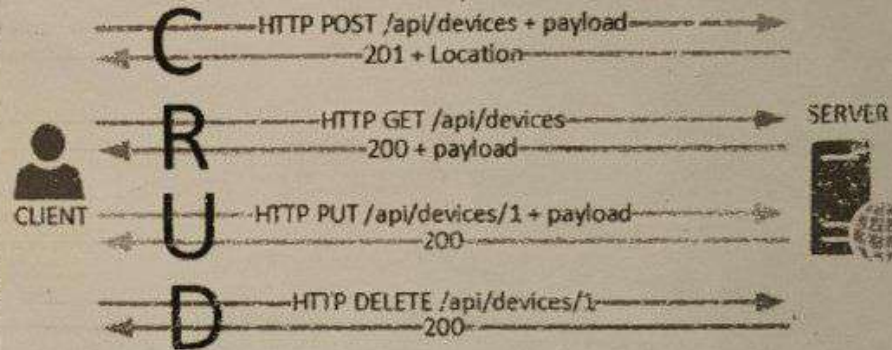
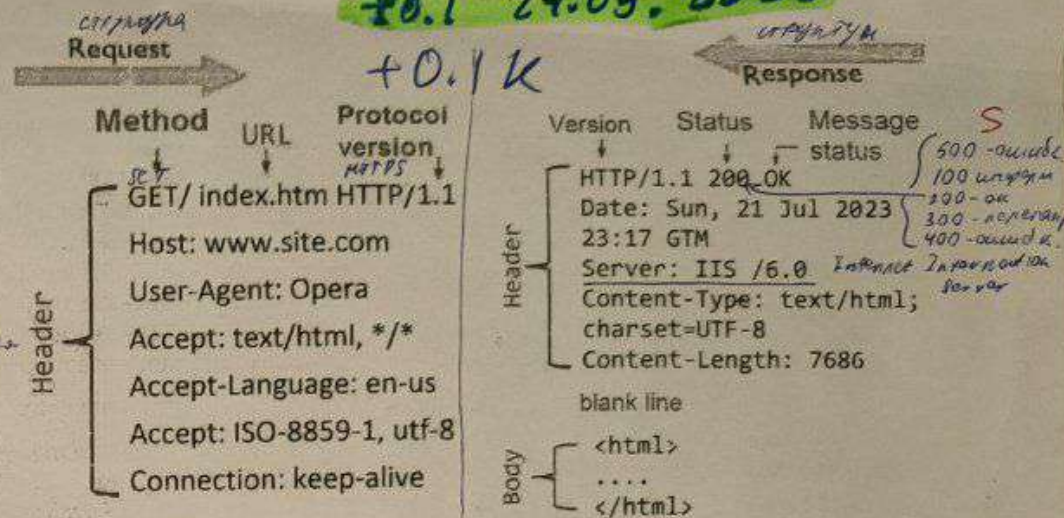
asp.net by/asperanto/1.htm - ludomik

+0.1 15.9.25

+0.1 18.9.25

+0.1 24.09.2025

+0.1 K

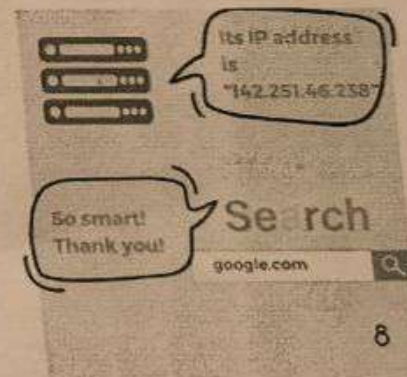
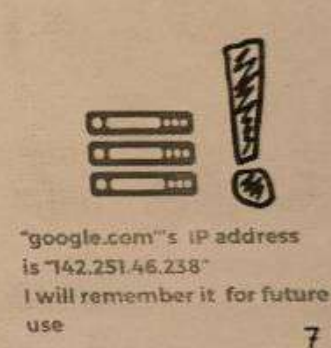
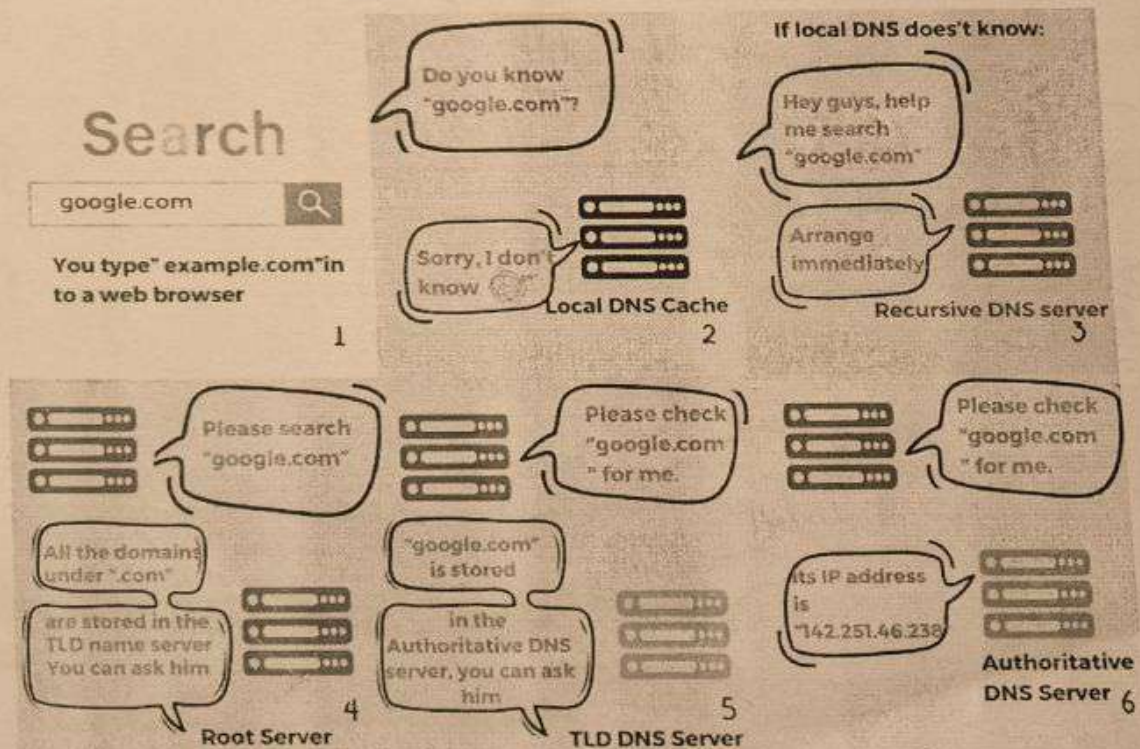
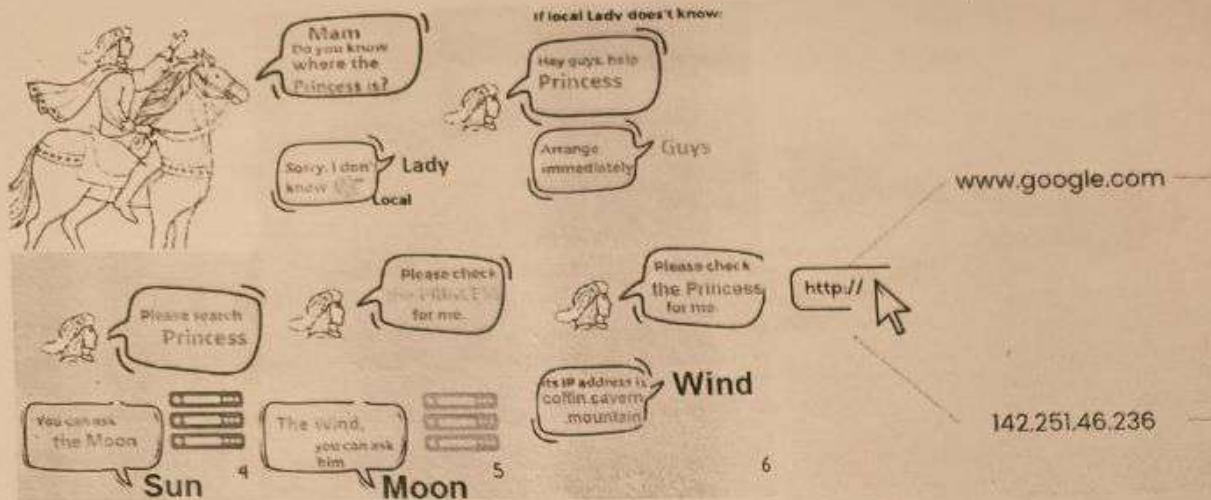


- GET — getting a resource
- POST — resource creation
- PUT — resource update
- DELETE — deleting a resource

Methods	Request		Response	
	URL	Request body	Status	Response body
1 GET	Yes	No	Yes	Yes
2 PUT	Yes	Yes	Yes	No
3 POST	Yes	Yes	Yes	Yes
4 DELETE	Yes	No	Yes	No

Create Read Update Delete

Чтение создание обновление удаление



```

class Acat {
    string name;
    public Acat(n) {
        this.name = n;
    }
}

```

osogorwin

Acat mycat:
new Acat("Boris")

polymorphism - разное поведение, при одинаковых условиях работы

В программировании - один объект генерирует разный результат

подана информация, и выводится результат

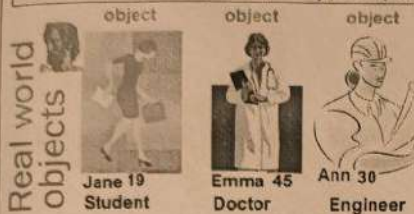
```

class ACat {
  string name;
  public ACat(n){
    this.name=n;
  }
}
ACat mycat= new ACat("Barsik");

```

extend - наследование

class To describe a Woman: name, age, job
Women can do: eat, drink, sleep, walk, ...



```

Say() {
  return "meou";
}
s = "My cat says <b>"
+ myCat.Say()+
"</b> !"

```

Error (+0.1 2015.2.27)

```

<script>
class APet { constructor(name)
  Say() {
    alert("No");
  }
}
class ACat extends APet {
  Say() {
    return "Miou";
  }
}
var myPet =
  new ACat("Barsik");

s = "My pet says <b>"
+ myPet.Say()+
"</b> !"
</script>

```

function WashDishes()

Men's version
wipe dry



Female version
wet with water



ira= new
AWoman()



ivan= new
AMan()

```

class AMan {
  WashDishes() {
    return 'wipe dry';
  }
}

```

```

class AWomen {
  WashDishes() {
    return 'wet';
  }
}

```

var family = [new AWomen(), new AMan()];
for(i=0;i<2;i++){ alert(" " + family[i].WashDishes());} //dry wet

women = new AWomen(); man = new AMan(); var family = [women,man];

ES6:

asp.net.by/es6/1.htm, asp.net.by/es6/02.htm

```
{
  let
  const
}
```

asp.net.by/es6/03.htm **Arrow functions:**

`x = (x, y) => y + ":" + x;`

`x("sbmt.by", "SBMT");`

asp.net.by/es6/04.htm function `myExam(y = 4)` -

параметр по умолчанию

... `myExam()`; - тогда выведет 4
... `myExam(5)`; - тогда выведет 5

asp.net.by/es6/05.htm Array.**find**(myFunction)

asp.net.by/es6/06.htm Array.**findIndex**()

asp.net.by/es6/07.htm Классы

class ACat {

constructor(n) {

классу принадлежит
`this.name = n;` //property

}

}

`mycat = new ACat("Барсик");`



class ACat{

Name()

return 'Меня зовут' + this.name;

}

`mycat = new ACat("Барсик");`

`document.write(mycat.Name());`

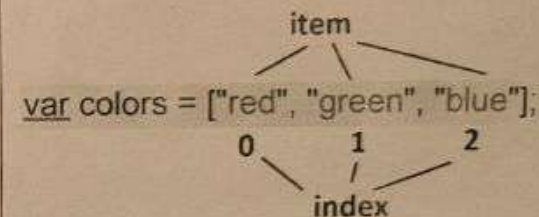
asp.net.by/es6/08.htm

Меня зовут Барсик

asp.net.by/es6/09.htm

Array.**forEach**()

`var colors = ["red",
"green", "blue"];`



`colors.forEach(myFunction);`

function **myFunction**(item, index) {

item ... index

}

Function One () {
// Do something
}

Function Two (call_One) {
// Do something else
call_One()
}

`Two(One);` ← code is being executed

Callback
illustrated

Promise

new Promise(executor)

state:
"pending"
result:
undefined

resolve(value)

reject(error)

state:
"fulfilled"
result:
value

state:
"rejected"
result:
error

ES6: event-driven programming

querySelector

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

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.

```
<html>
<head>
  <title>Hello</title>
</head>
<body>
  <h2></h2>
  <button onclick="Hello()">Click Me!</button>
</body>
</html>
```

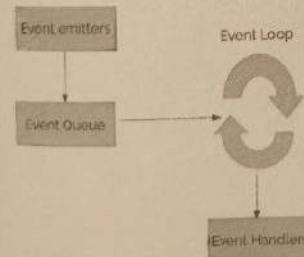
```
document.addEventListener(
  DOMContentLoaded, function(){
    document.querySelector("button").onclick = Hello;
  });
```

```
<html>
<head>
  <title>Hello</title>
</head>
<body>
  <h2></h2>
```

```
<script>
  document.querySelector("button").onclick = Hello;
</script>
<button onclick="Hello()">Click Me!</button>
<script>
  document.querySelector("button").onclick = Hello;
</script>
</body>
</html>
```

Uncaught TypeError: Cannot set properties of null (setting 'onclick') at counter.html:13:45

document.querySelector("button").addEventListener('click', Hello);



```
<button onclick="Hello()">Click Here</button>
<script>
  let counter = 0;
  function Hello(){
    counter++;
    const heading = document.querySelector('h1');
    heading.innerHTML = counter;
  }
</script>
```

as you can see as you can see the script is not executed until the document is loaded

DOM - Document Object Model

F12 - console.log

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>
```

Name

Отправить

input field where the user can type in some text

```
<script>
  document.addEventListener('DOMContentLoaded', function() {
    document.querySelector('form').onsubmit = function() {
      let name = document.querySelector('#name').value;
      alert('Hello, ${name}!');
    };
  });
</script>
```

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

https://bbl1247.w3spaces.com/coloc1.html

Инспектор Консоль Отладчик Ссылки () Справка

```
>> 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(), ... }
```

```
document.addEventListener('DOMContentLoaded', function(){
  document.querySelector('button').onclick = Hello; }
});
document.querySelector('#red').onclick = function() {
  document.querySelector('#hello').style.color = 'red'; };
```

using JS events:

① inline HTML event handlers

② add event listener()

③ Traditional DOM event handler