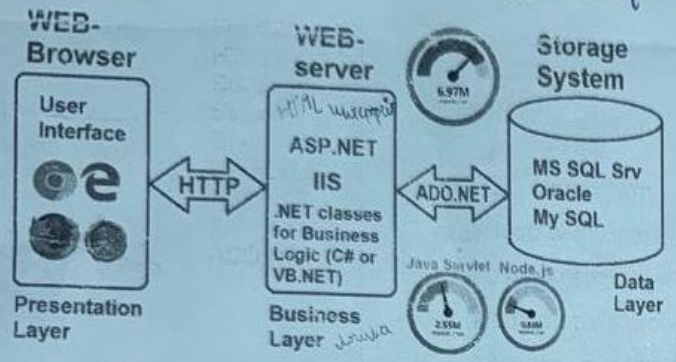
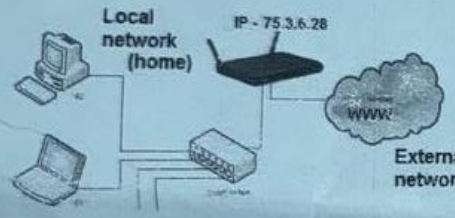
+0,3 to fin Grade public ISB

+0.3 to fin Grade 11.9.

(5) (@) → Новые почты инф 1,2,3



Front End
HTML/ CSS/ Java Script и DOM
BackEnd
ASP.NET
IIS
ADO.NET



DHCP (engl. Dynamic Host Confi-guration Protocol)

tracert - traceroute (in Apple) www.geotraceroute.com

172 . 16 . 254 . 1

10101100.0001 0000.11111110.00000001

8 bits

32 bits (4 bytes)

4 billion addresses – 4.294.967.296

DNS – clearer address.

217.21.43.40 – sbmt.by, sbmt.bsu.by sb.bsu.by

ping sb.bsu.by tracert sb.bsu.by https://geotraceroute.com/

(1) (2) (3)

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

где сделать каталог

1 IP – phylantr

3. Город, где ваш сайт есть

4. philantropy.bsite.net/WEB

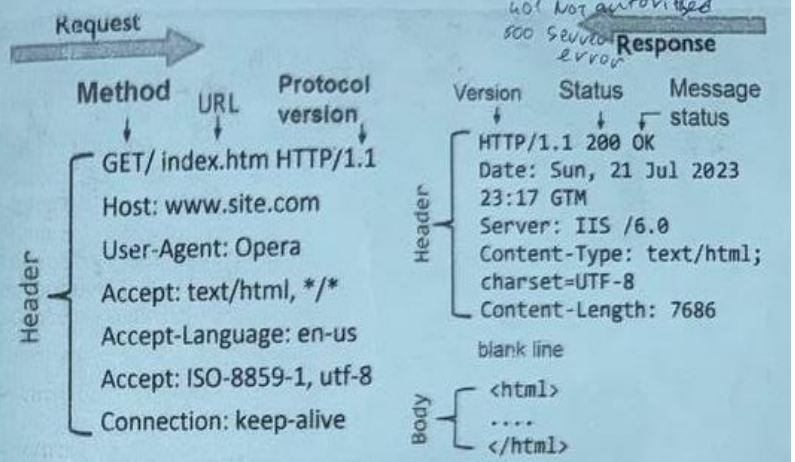
bdy – увеличение курса ↑

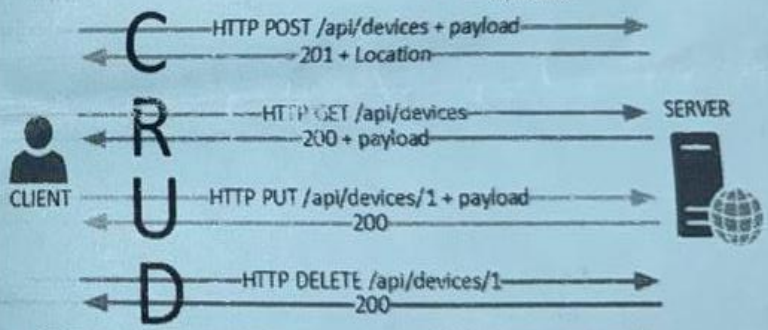
bear – понижение курса ↓

+0.1 2025.9.24

+0.1 15.9.25 +0.1 18.9.25

Status: 100 продолжение
200 OK
301 redirection
404 Not found
401 Not authorized
500 Service error





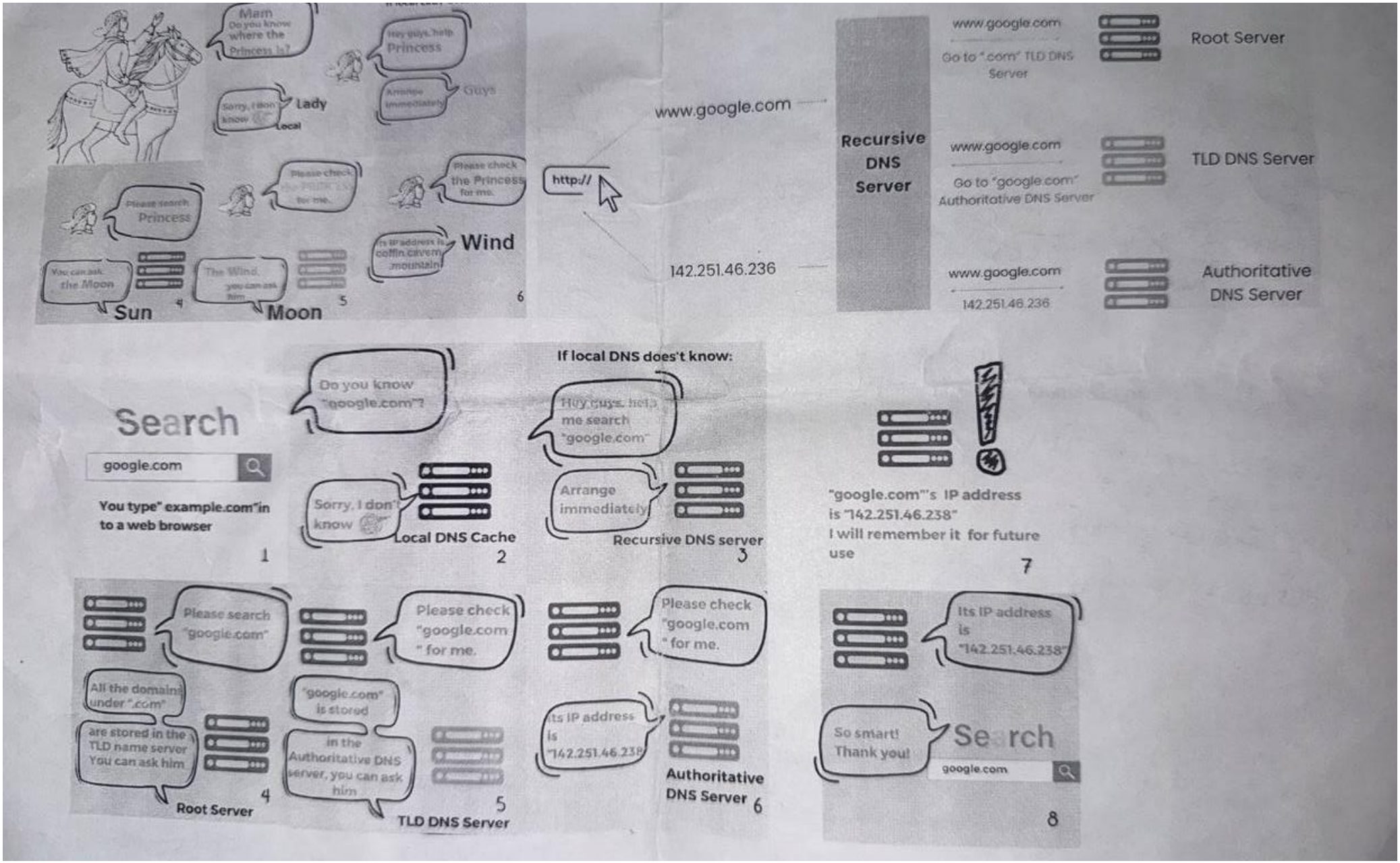
- GET -- getting a resource
- POST — resource creation
- PUT — resource update
- DELETE — deleting a resource

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

Webserver – IIS это windows

Apache Tomcat это Linux

Samee Code


Man
Do you know where the Princess is?
Hey guys, help Princess
Arrange immediately
Guys
Sorry, I don't know
Local
Lady
Please search Princess
Please check
Please check the Princess for me.
Its IP address is coffin,cavern, mountain
Wind
You can ask the Moon
Sun
4
The Wind, you can ask him
Moon
5
6
http://
www.google.com
142.251.46.236
Recursive DNS Server
www.google.com
Go to ".com" TLD DNS Server
Root Server
www.google.com
Go to "google.com" Authoritative DNS Server
TLD DNS Server
www.google.com
142.251.46.236
Authoritative DNS Server
Search
google.com
You type" example.com"in to a web browser
1
Do you know "google.com"?
Sorry, I don't know
Local DNS Cache
2
If local DNS does't know:
Hey Guys, help me search "google.com"
Arrange immediately
Recursive DNS server
3
"google.com"'s IP address is "142.251.46.238"
I will remember it for future use
7
Please search "google.com"
All the domains under ".com" are stored in the TLD name server You can ask him
Root Server
4
Please check "google.com " for me.
"google.com" is stored in the Authoritative DNS server, you can ask him
TLD DNS Server
5
Please check "google.com " for me.
Its IP address is "142.251.46.238"
Authoritative DNS Server
6
Its IP address is "142.251.46.238"
So smart! Thank you!
Search
google.com
8

philantropy.bsite.net/Project/1/

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="robots" content="noindex">
```


```
<style>
        #container {
          display: flex;
          flex-wrap: wrap;
          align: center;
        }
        #container > div {
          background-color: gray;
          font-size: 20px;
          margin: 20px;
          padding: 20px;
          width: 200px;
          text-align: center;
        }
</style>
```

```
<body>
.....
    <div id="container">
      <div>Project 0</a></div>
      <div>Project 1</div>
      <div>Project 2</div>
      <div>Project 3</div>
    </div>

</body>
<html>
```

Ludovik Zamenhof

@media

```
<!-- Google Font -->
  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;700&display=swap"
rel="stylesheet">
  <style>
    body {
      font-family: 'Poppins', sans-serif;
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <div class="buttons">
    <a href="project1.html" class="btn btn-primary btn-project">Project 1</a>
```

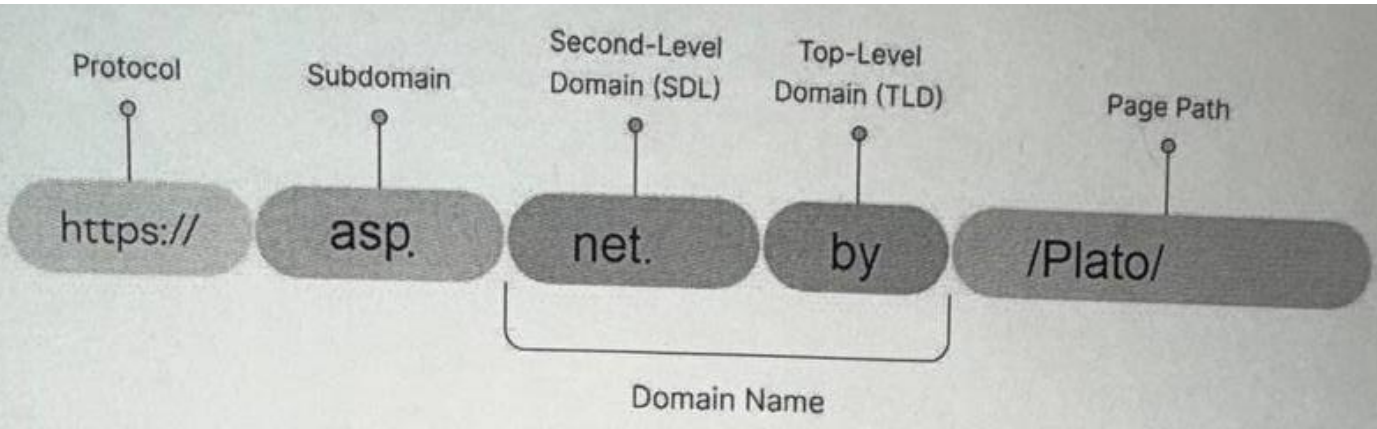
```
@media (min-width: 768px) {
    .profile-image {
      width: 200px;
      height: 200px;
    }
    .name {
      font-size: 2.5rem;
    }
}
```

to enlarge photos on large screens from 150 to 200px

The same goes for fonts

philantropy.bsite.net/WEB-T/~~Proj~~ — Projects

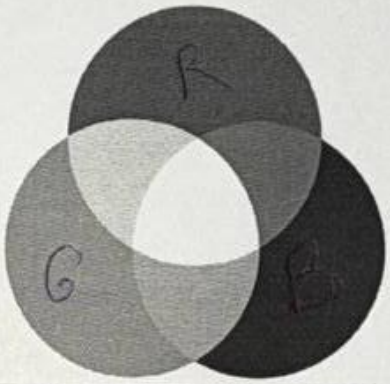
babok.bsite.net/babko/ Project 2 /1/ /2/ — Bootstrap

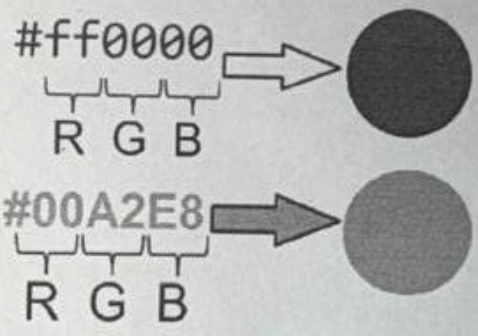


R- Red

G- Green

B – Blue





Intensity

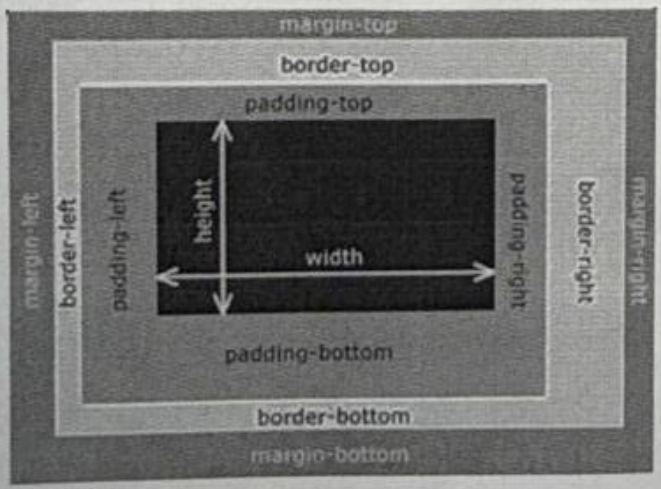
0	1	1	0	1	0	0	1

8-digit cell

rgb(255,0,0), **rgb(0,0,255)**, **rgb(0,162,232)**, **rgb(31,78,121)**

rgb(100%,0%,0%), **rgb(0%,0%,100%)**

Box model




```
<style>
div
{
 color:white;
 background-color:brown;
 height: 250px; width: 100px;
 padding: 20px 11px 10px 15px;
 border: 15px solid yellow;
 margin: 120px 10px 15px 20px;
}
</style>

<div><b>Box Model</b><br>
```

```
<style>
img {
  position: absolute;
  left: 2px;
  top: 10px;
  z-index: -1; // behind the text }
</style>
```

asp.net.by/Bootstrap/
getbootstrap.com

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

передача функции

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) {
    this.name = n;    //property
  }
}
```

это без this будет undefined

```
mycat = new ACat("Барсик");
```

```
class ACat{
.....
  Name()
    return 'Меня зовут ' + this.name;
  }
}
mycat = new ACat("Барсик");
document.write(mycat.Name());
```

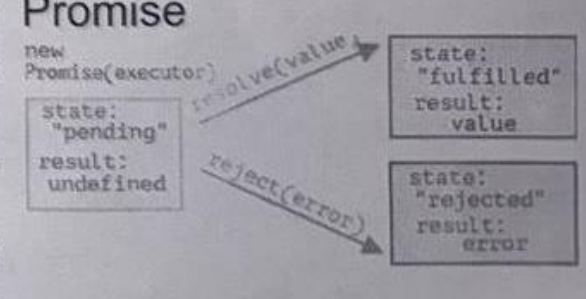
asp.net.by/es6/08.htm asp.net.by/es6/09.htm Array.**forEach**() var colors = ["red", "green", "blue"];	Меня зовут Барсик item var colors = ["red", "green", "blue"]; 0 1 2 index

```
colors.forEach(myFunction);
function myFunction(item, index) {
  item … index
}
```

Callback illustrated

```
Function One (){
    // Do something
}
Function Two (call_One){
    // Do something else
    call_One()
}
Two(One);
```

code is being executed



Error

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

```
class ACat {
 constructor(n) {
  this.name = n;  //property
 }
}
mycat = new ACat("Barsik");
```

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

Error

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

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

наследование

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

Real world objects

object
Jane 19
Student

object
Emma 45
Doctor

object
Ann 30
Engineer

Polymorphism in Biology

APerson

AWoman extends APerson

AMan extends APerson

function **WashDishes()**

Men's version
wipe dry

Female version
wet with water

ira= new AWoman()

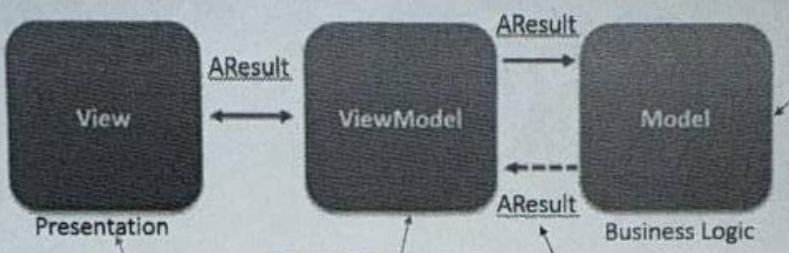
ivan= new AMan()

```
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];
```

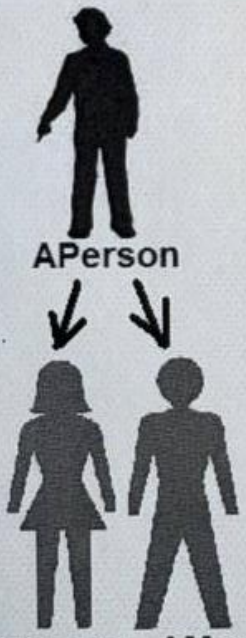



```
AResult result =
    Model();
View(result);

Console.ReadKey();
```

```
class AResult {
    internal string single ="";
    internal string family ="";
}
```

```
static void View( AResult result) {
  Console.WriteLine("Version for singles: \n\r "+result.single+"\n\r ");
  Console.WriteLine("Version for families: \n\r " + result.family + "\n\r ");
}
```

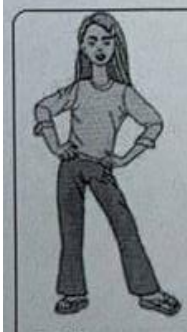
```
static AResult Model() {
    AResult result = new AResult();
    var single = new APerson();
    result.single += single.GetType().Name + ": " +
                        single.WashDishes() + "";
    APerson woman = new AWomen();
    APerson man = new AMan();
    APerson[] family = new APerson[2];
    family[0] = woman;
    family[1] = man;
    for (int i = 0; i < 2; i++)
    {
      result.family += family[i].GetType().Name + ": " +
            family[i].WashDishes() + "\n\r ";
    }
    return result;
}
```



Men's version
wipe dry

Female version
wet with water

```
class APerson {
    virtual public string WashDishes() {
        return "dry and wet";
    }
```

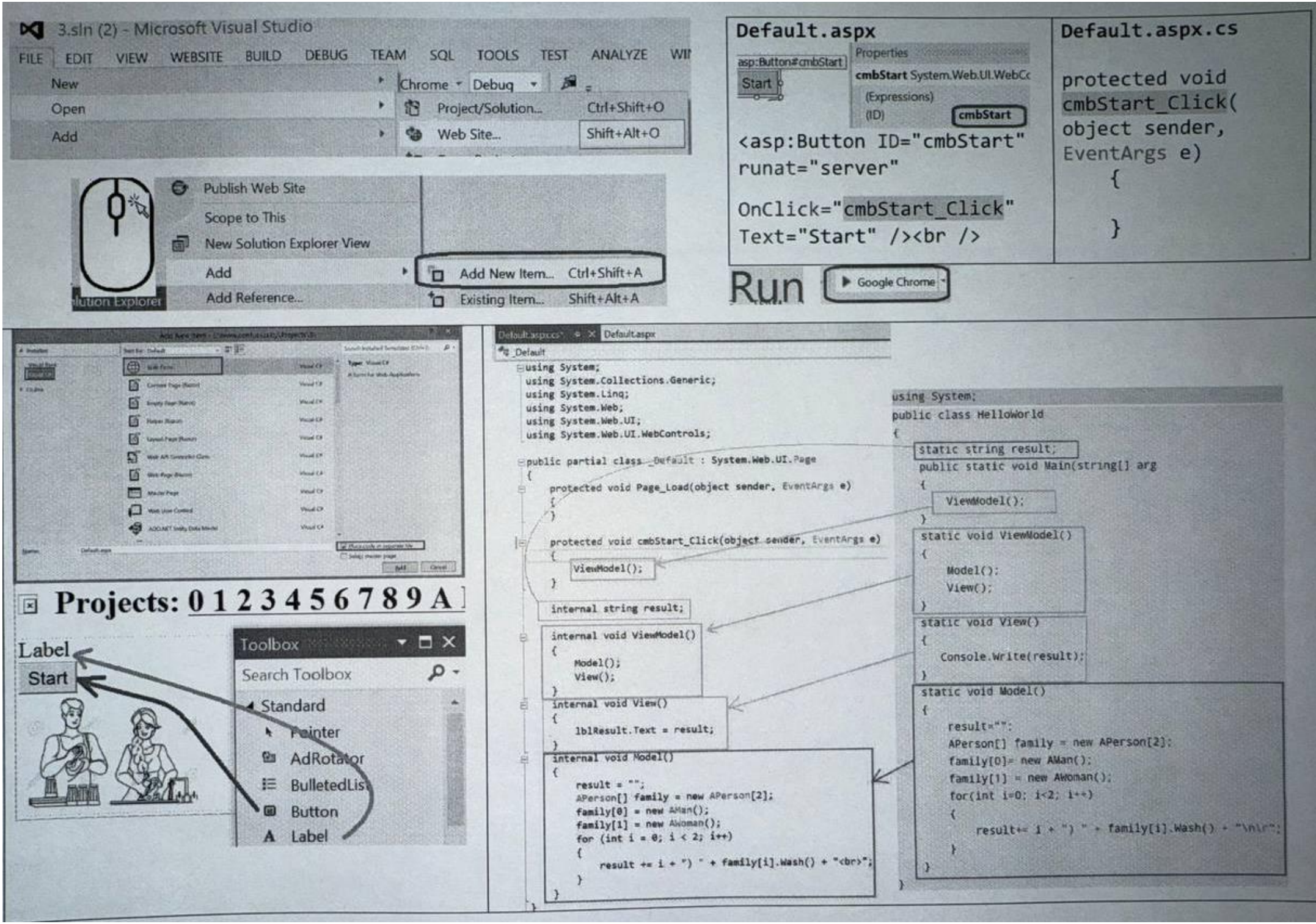


AWoman
ira= new
AWoman()

AMan
ivan= new
AMan()

```
class AMan: APerson {
override public
   string WashDishes()
{
    return "dry";
  }
}
```

```
class AWoman: APerson {
 override  public
      string WashDishes()
{
    return "wet";
  }
}
```


3.sln (2) - Microsoft Visual Studio
FILE EDIT VIEW WEBSITE BUILD DEBUG TEAM SQL TOOLS TEST ANALYZE
New
Open
Add
Chrome Debug
Project/Solution... Ctrl+Shift+O
Web Site... Shift+Alt+O
Publish Web Site
Scope to This
New Solution Explorer View
Add
Add Reference...
Add New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
Default.aspx
asp:Button#cmbStart
Start
Properties
cmbStart System.Web.UI.WebC
(Expressions)
(ID) cmbStart
<asp:Button ID="cmbStart"
runat="server"
OnClick="cmbStart_Click"
Text="Start" />
Default.aspx.cs
protected void
cmbStart_Click(
object sender,
EventArgs e)
{
}
Run
Google Chrome
Projects: 0 1 2 3 4 5 6 7 8 9 A
Label
Start
Toolbox
Search Toolbox
Standard
Pointer
AdRotator
BulletedList
Button
Label
Default.aspx.cs
Default.aspx
_Default
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
}
protected void cmbStart_Click(object sender, EventArgs e)
{
ViewModel();
}
internal string result;
internal void ViewModel()
{
Model();
View();
}
internal void View()
{
lblResult.Text = result;
}
internal void Model()
{
result = "";
APerson[] family = new APerson[2];
family[0] = new AMan();
family[1] = new AWoman();
for (int i = 0; i < 2; i++)
{
result += i + ") " + family[i].Wash() + "
";
}
}
}
using System;
public class HelloWorld
{
static string result;
public static void Main(string[] arg
{
ViewModel();
}
static void ViewModel()
{
Model();
View();
}
static void View()
{
Console.Write(result);
}
static void Model()
{
result="";
APerson[] family = new APerson[2];
family[0]= new AMan();
family[1] = new AWoman();
for(int i=0; i<2; i++)
{
result+= i + ") " + family[i].Wash() + "\n\r";
}
}

```
class APerson
{
    virtual internal string Wash()
    {
        return "Wet and Dry";
    }
}
```

```
class AMan:APerson
{
    override internal string Wash()
    {
        return "Wet";
    }
}
```

```
class AWoman:APerson
{
    override internal string Wash()
    {
        return "Dry";
    }
}
```


```
class APerson
{
    virtual internal string Wash()
    {
        return "Wet and Dry";
    }
}
```

```
class AMan:APerson
{
    override internal string Wash()
    {
        return "Wet";
    }
}
```

```
class AWoman:APerson
{
    override internal string Wash()
    {
        return "Dry";
    }
}
```


ES6: event-driven programming

querySelector

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

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

selects the first HTML element which the query applies to

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

variable defination, we could use let or var instead. But const id 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.

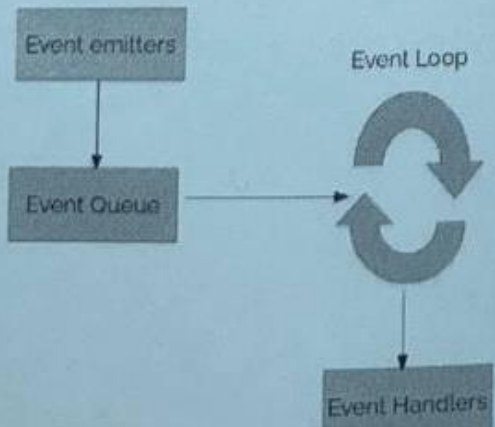
```
<html>
<head>
  <title>Hello</title>
  <script>
  function Hello(){
document.querySelector('h2').innerHTML=
                    "Hello JavaScript!";}
  </script>
</head>
<body>
<h2></h2>
<button onclick="Hello()">Click
Me!</button>
</body>
</html>
```

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

```
<html>
<head>
  <title>Hello</title>
</head>
  <script>
  function Hello(){
document.querySelector('h2').innerHTML="Hello JS!";}
  </script>
 <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  counter.htm:13
TypeError: Cannot set
properties of null (setting
'onclick')
    at counter.htm: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>
```

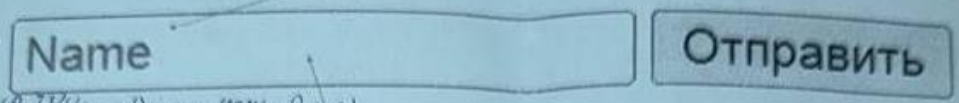
HTML forms

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>
```

сначала обязан id, затем <script>

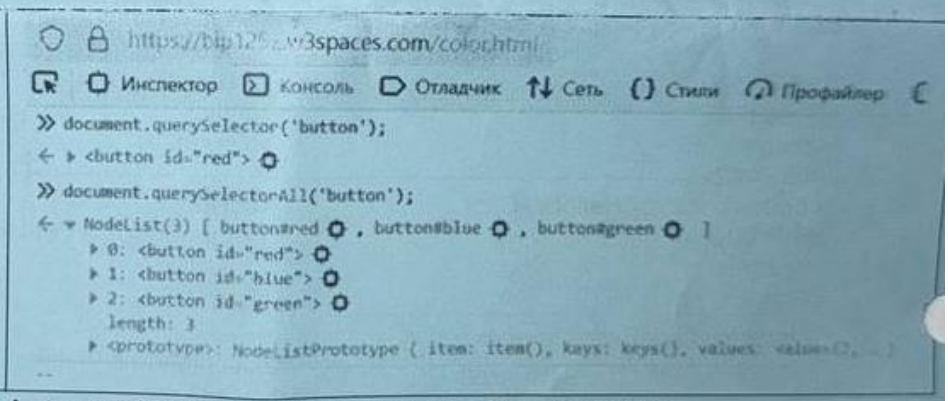



```
<p id='demo'></p>
<script>
document.getElementById('demo').innerHTML = Date();
</script>
```

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>
```



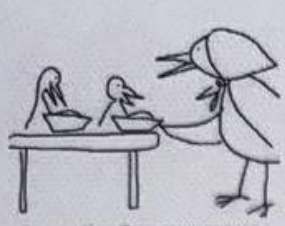

```
document.addEventListener('DOMContentLoaded',function(){
document.querySelector('button').onclick = Hello; }
);
document.querySelector('#red').onclick = function() {
   document.querySelector('#hello').style.color = 'red'; };
```

ways of using JS events → Traditional DOM Event Handler

inline HTML event handlers

addEventListener()

```
<script>
alert("Hi");
</script>
```



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

querySelectorAll() ⇨ [,]
NodeList

[,].forEach( );

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

Red Blue Green

When the page is done loading

```
<script>
    document.addEventListener('DOMContentLoaded', function() {
        document.querySelectorAll('.color-change').forEach(
            function(button) {
button.onclick = function() {
document.querySelector('#hello').style.color = button.dataset.color;
            };
        });
    });
</script>
```

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

looked for things of the particular class

HTML

```
<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>
```

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

```
function(button) {
    button.onclick = function() {

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

```
<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>
```

Hello!
Red
Black
Red
Blue
Green

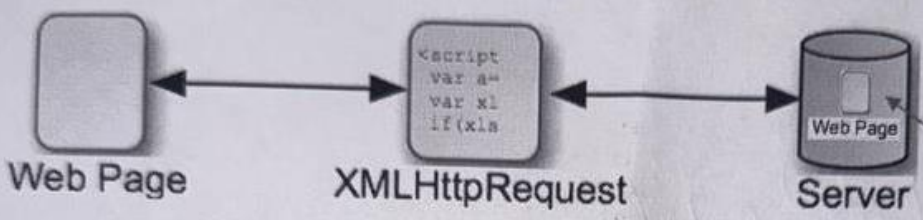
```
function() {...}

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

() => {...}

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

bpmn2.bsite.net/A/Projects/UC/1.htm

```
{"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>
```



For security reasons, modern browsers do not allow access across domains.

All must be located on the same server.

Browser

https://asp.net.by/ajax/ajax_info.txt

Server

An event occurs...

- Create an XMLHttpRequest object
- Send HttpRequest

Internet

- Process HTTPRequest
- Create a response and send data back to the browser

XMLHttpRequest

onreadystatechange = function() { ... };
readyState = DONE
status = 200
statusText = "OK"
open() send()
responseText = { ... }

The response value from the server is assigned to the **responseText** property

```
<!DOCTYPE html>
<html>
<body>
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange =
   function() {
    if (this.readyState == 4 && this.status == 200) {
document.getElementById("demo").innerHTML =
                                this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

<div id="demo">
  <h2>Let AJAX change this text.</h2>
  <button type="button" onclick="loadDoc()">GO
                                    </button>
</div>
```

Browser

- Process the returned data using JavaScript
- Update page content

bpmn2.bsite.net/A/Projects/UC/1.htm
2.htm