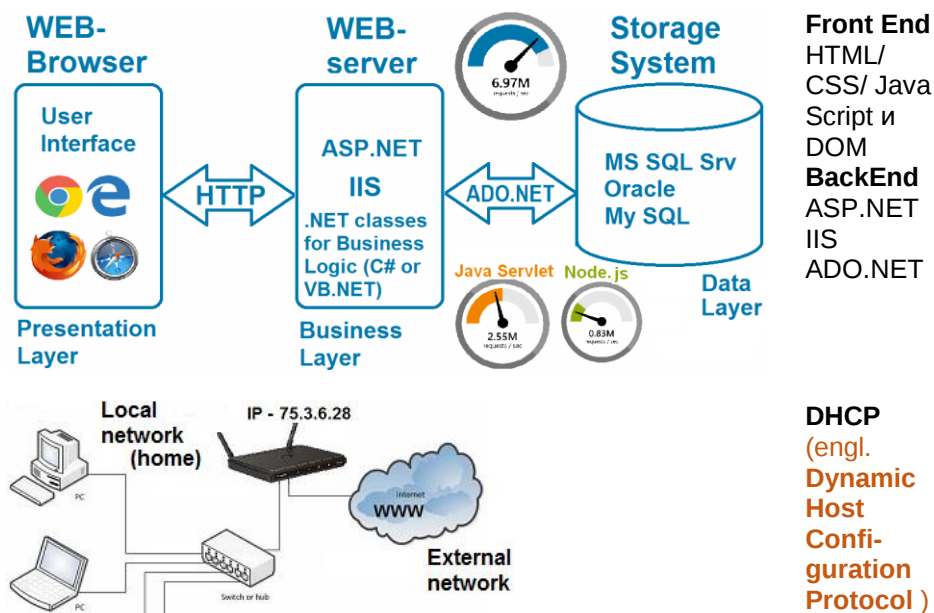
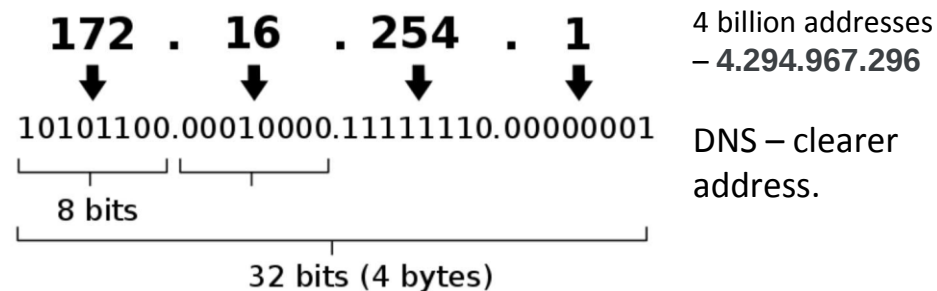


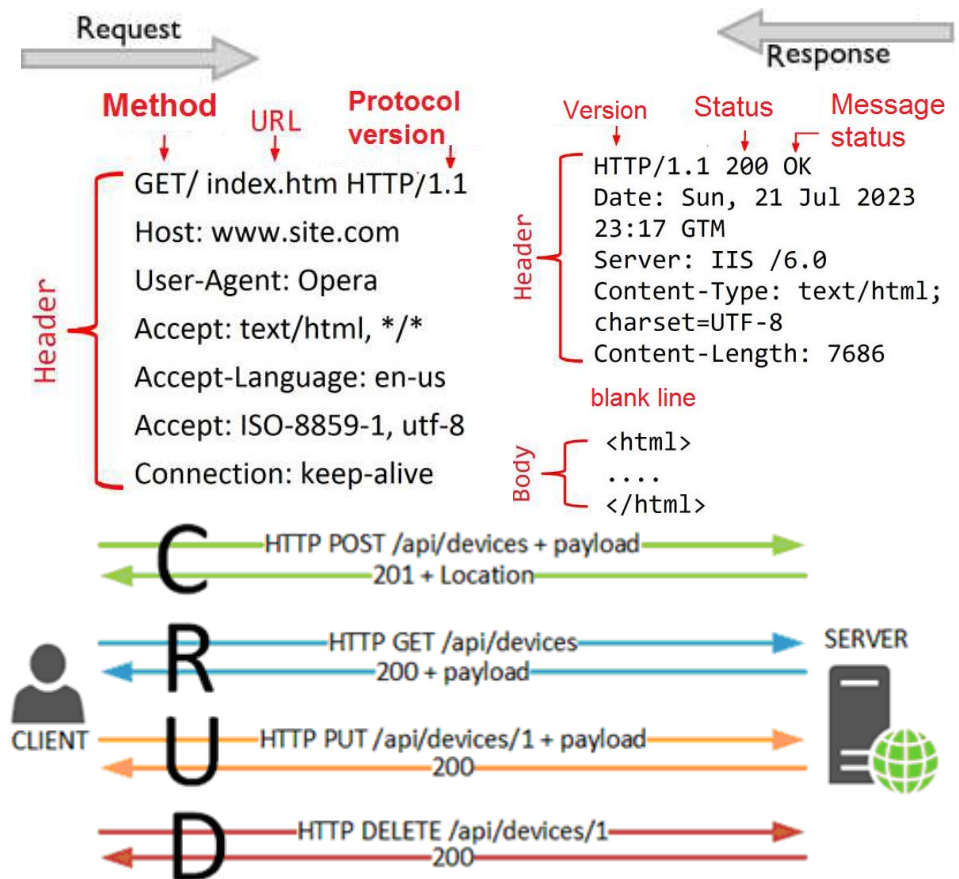


tracert – traceroute (in Apple) www.geotracerroute.com



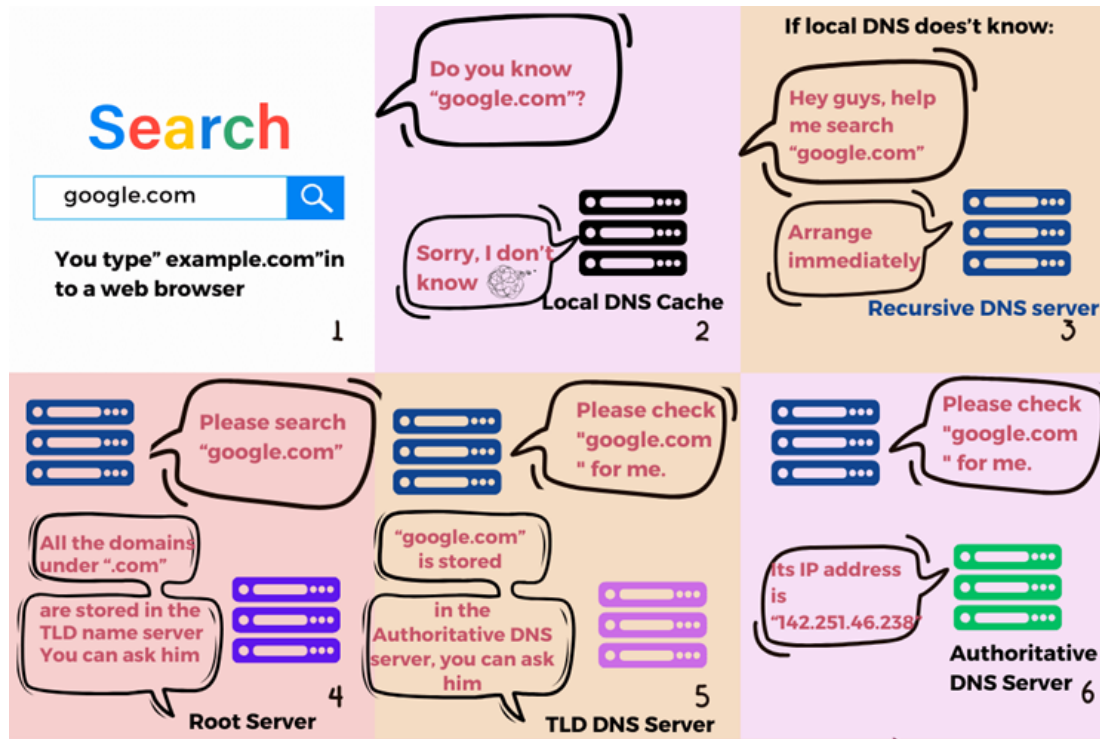
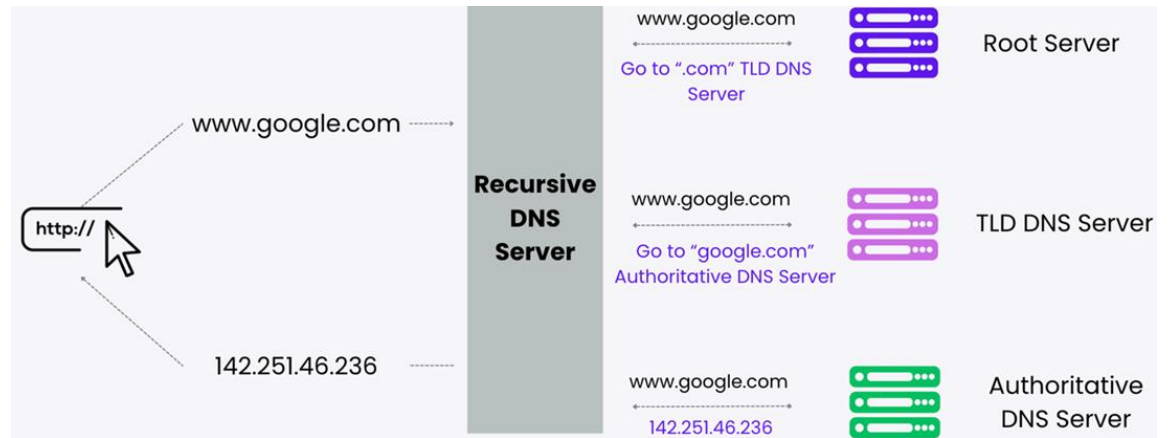
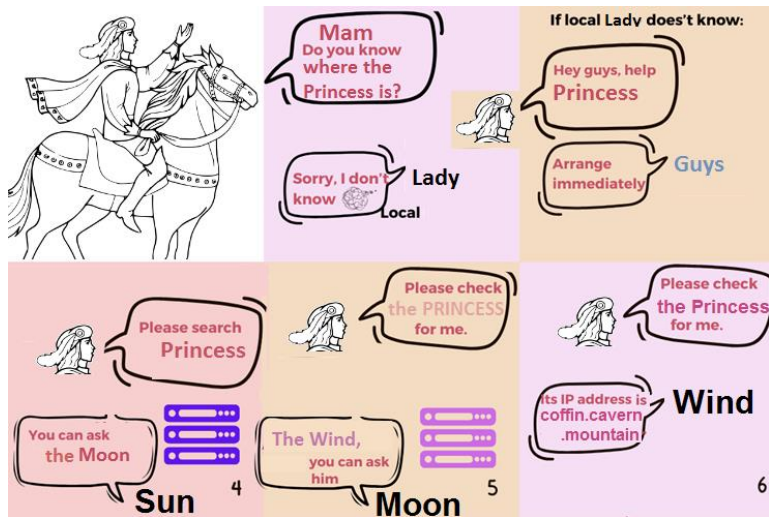
217.21.43.40 – sbmt.by, sbmt.bsu.by sb.bsu.by
ping sb.bsu.by tracert sb.bsu.by <https://geotracerroute.com/>

+375 (29) 254 07 92 - [ANDREY O. YAROSHEVICH](mailto:ANDREY.O.YAROSHEVICH)



- 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



```

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

```



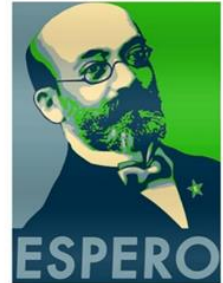
```

<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</div>
    <div>Project 1</div>
    <div>Project 2</div>
    <div>Project 3</div>
  </div>
</body>
<html>

```

Ludovik Zamenhof



Project 0

Project 1

Project 2

Project 3

<https://dot.net.by/esperanto/>



```

<!-- 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;

```

```

    }

```

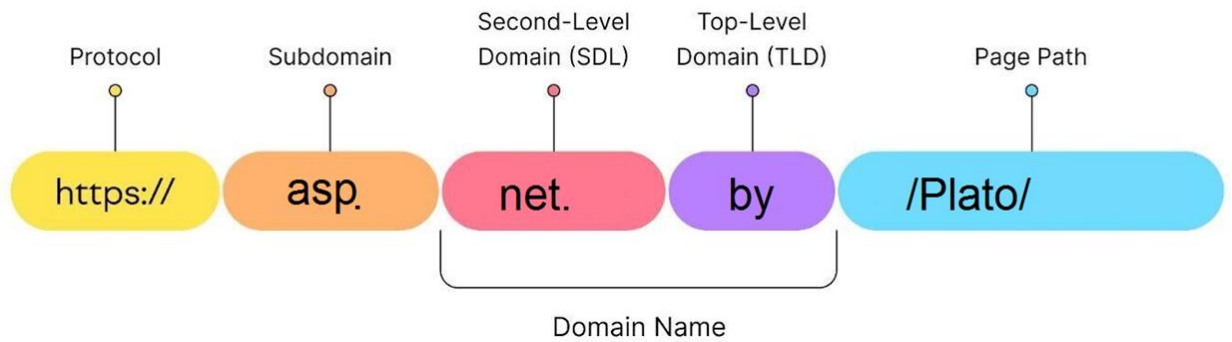
```

  }

```

to enlarge photos on large screens from 150 to 200px

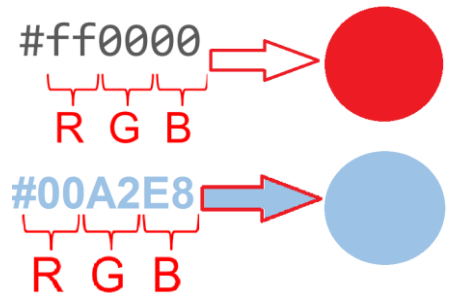
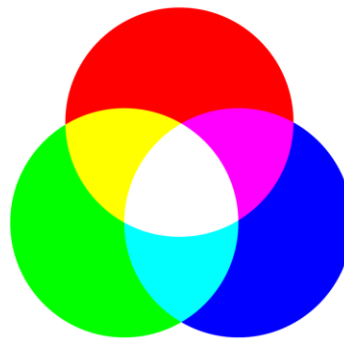
The same goes for fonts



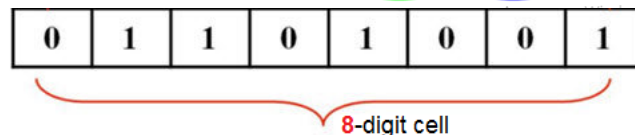
R- Red

G- Green

B – Blue



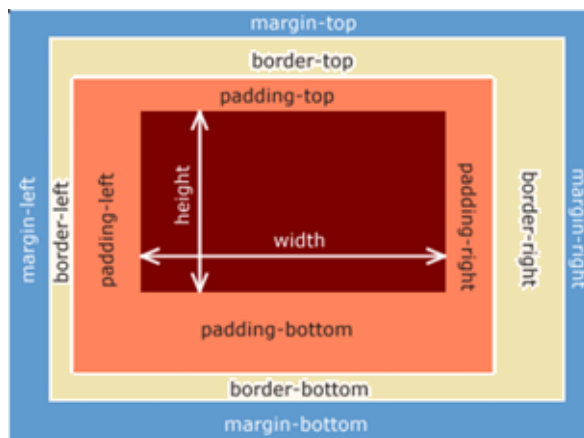
Intensity



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>Box Model


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



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

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  
  
  }  
  
}
```



```
mycat = new ACat("Барсик");
```

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

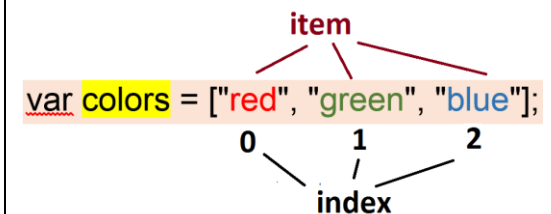
asp.net.by/es6/08.htm

asp.net.by/es6/09.htm

Array.**forEach**()

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

Меня зовут Барсик



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

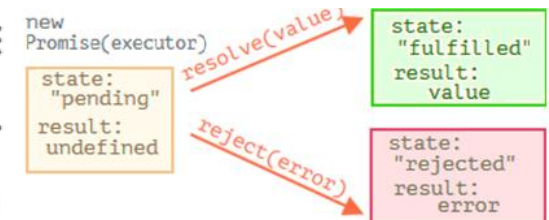
```
Function One (){  
  // Do something  
}  
  
Function Two (call_One){  
  // Do something else  
  call_One()  
}
```

```
Two(One);
```

code is being executed

Callback
illustrated

Promise



```
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> !"

```

```
<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>
```

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

Real world
objects



object



Jane 19
Student

object



Emma 45
Doctor

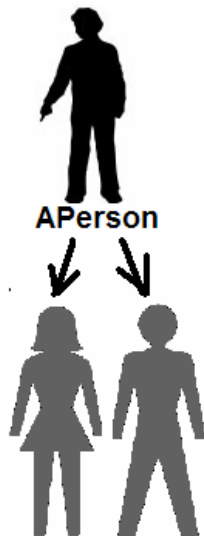
object



Ann 30
Engineer



Polymorphism in Biology



AWoman
extends
APerson

AMan
extends
APerson

function WashDishes()

Men's version
wipe dry



Female version
wet with water



ira= new
AWoman()



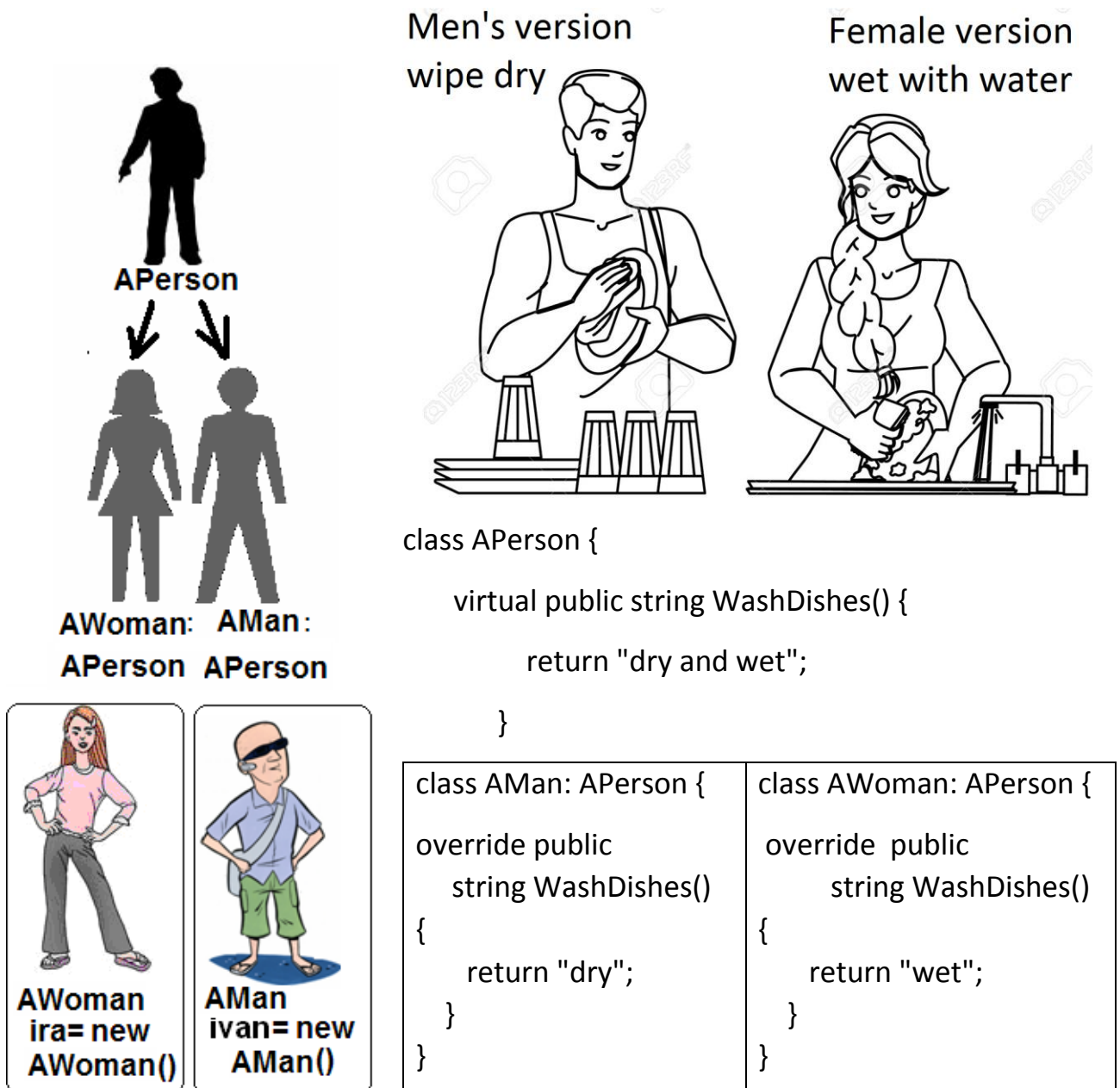
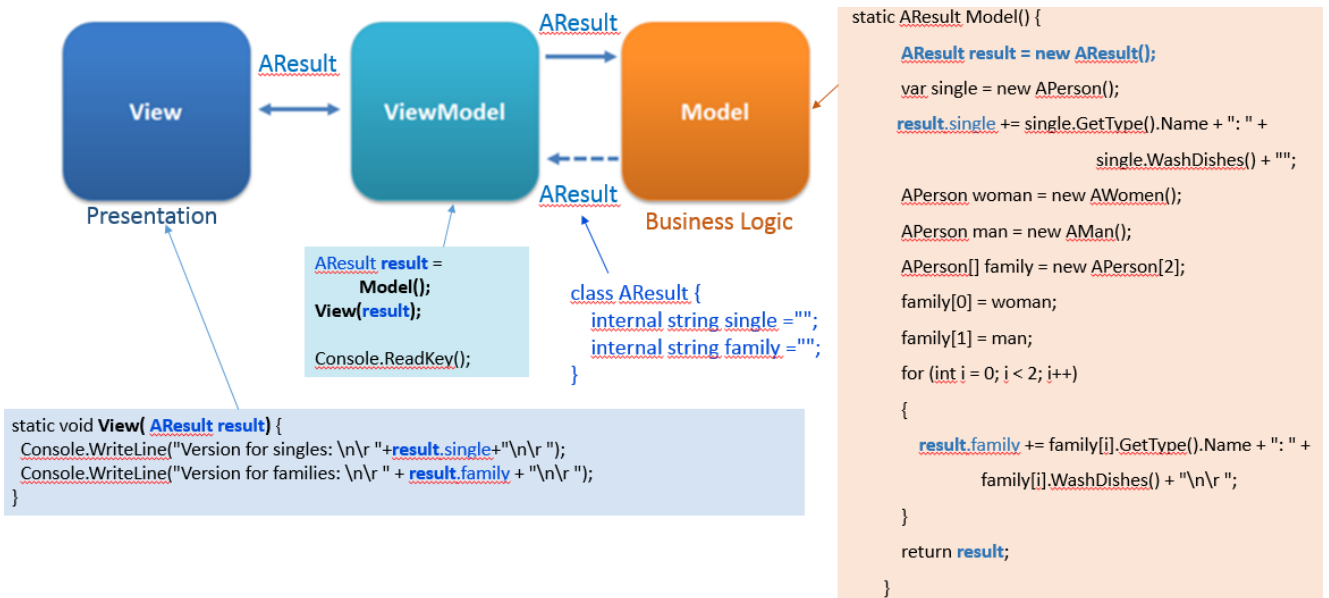
ivan= new
AMan()

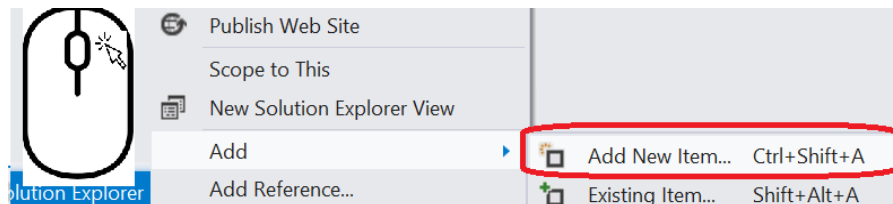
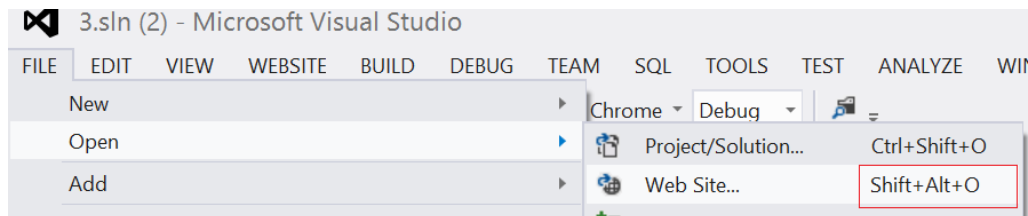
```
class AMan {
  WashDishes() {
    return 'wipe dry';
  }
}
```

```
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];
```





Default.aspx

```

asp:Button#cmbStart
Start
<asp:Button ID="cmbStart"
runat="server"
OnClick="cmbStart_Click"
Text="Start" /><br />

```

Default.aspx.cs

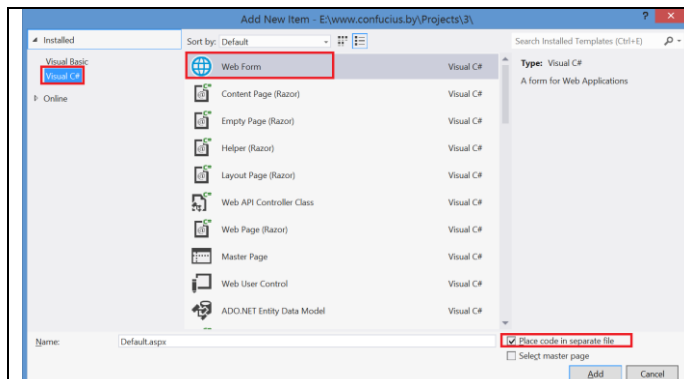
```

protected void
cmbStart_Click(
object sender,
EventArgs e)
{
}

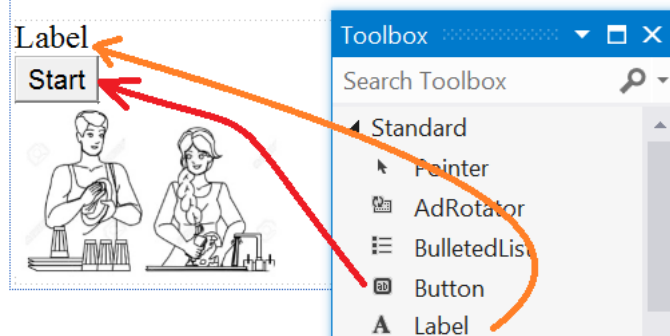
```

Run

Google Chrome



Projects: 0 1 2 3 4 5 6 7 8 9 A



Default.aspx.cs

```

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() + "<br>";
        }
    }
}

```

Default.aspx

```

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";
    }
}

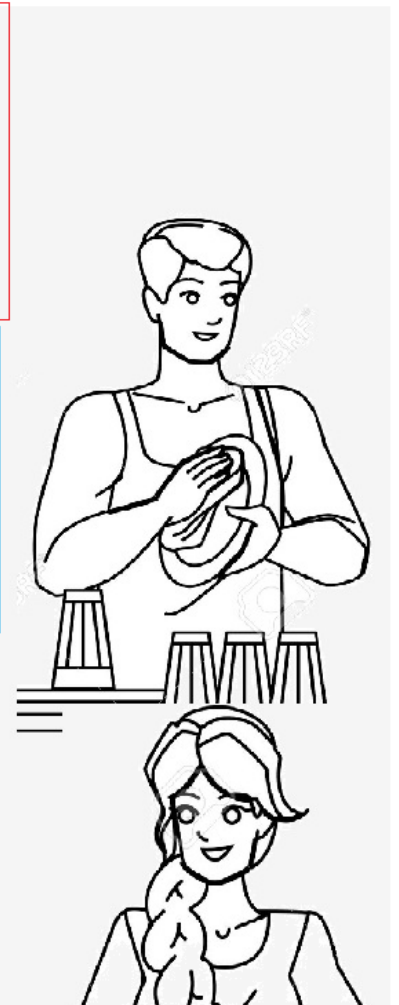
class AMan:APerson
{
    override internal string Wash()
    {
        return "Wet";
    }
}

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

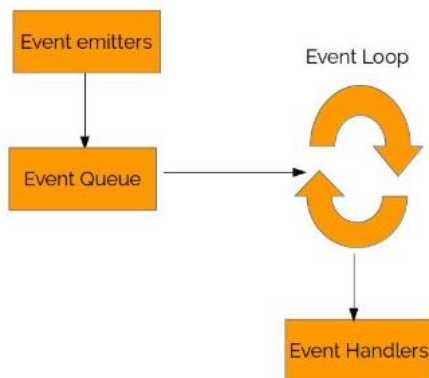
```
class APerson
{
    virtual internal string Wash()
    {
        return "Wet and Dry";
    }
}

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

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



ES6: event-driven programming



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

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.

<pre><button id="red">Red</button> <button id="blue">Blue</button> <button id="green">Green</button></pre>	<pre>>> 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(), ... }</pre>
--	--

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

```

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

```

<=JSON vs XML=>

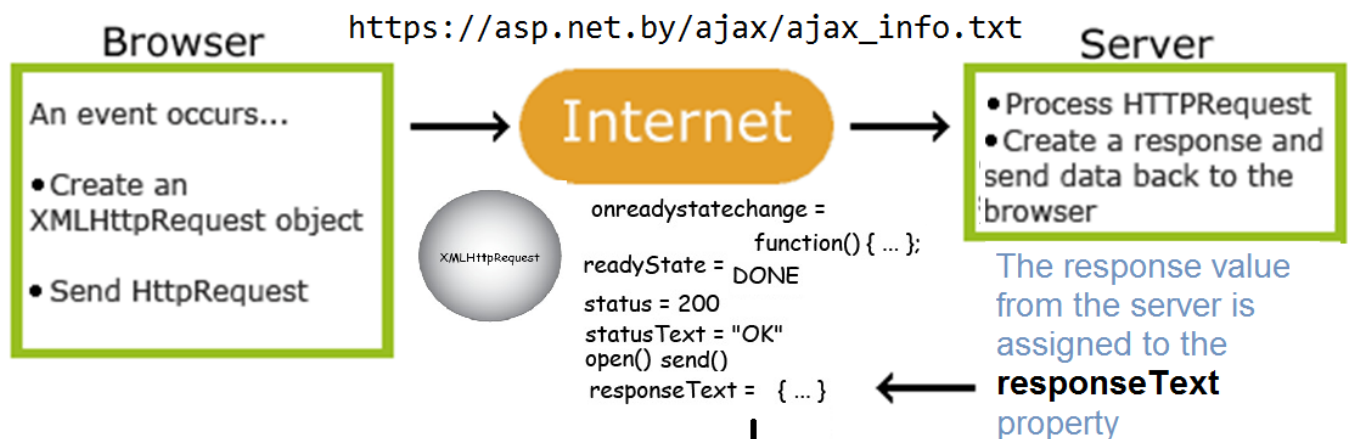
<pre> <manager> <firstName>Bill</firstName> <lastName>Gates</lastName> </manager> </pre>	
<pre> <manager> <firstName>Mike</firstName> <lastName>Dell</lastName> </manager> </pre>	

</managers>



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

All must be located on the same server.



```

<!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



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

```
<script>
  When the page is done loading
  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>
```

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 then the second button then the third button

```
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>
```



```
function() {...}
```

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

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

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