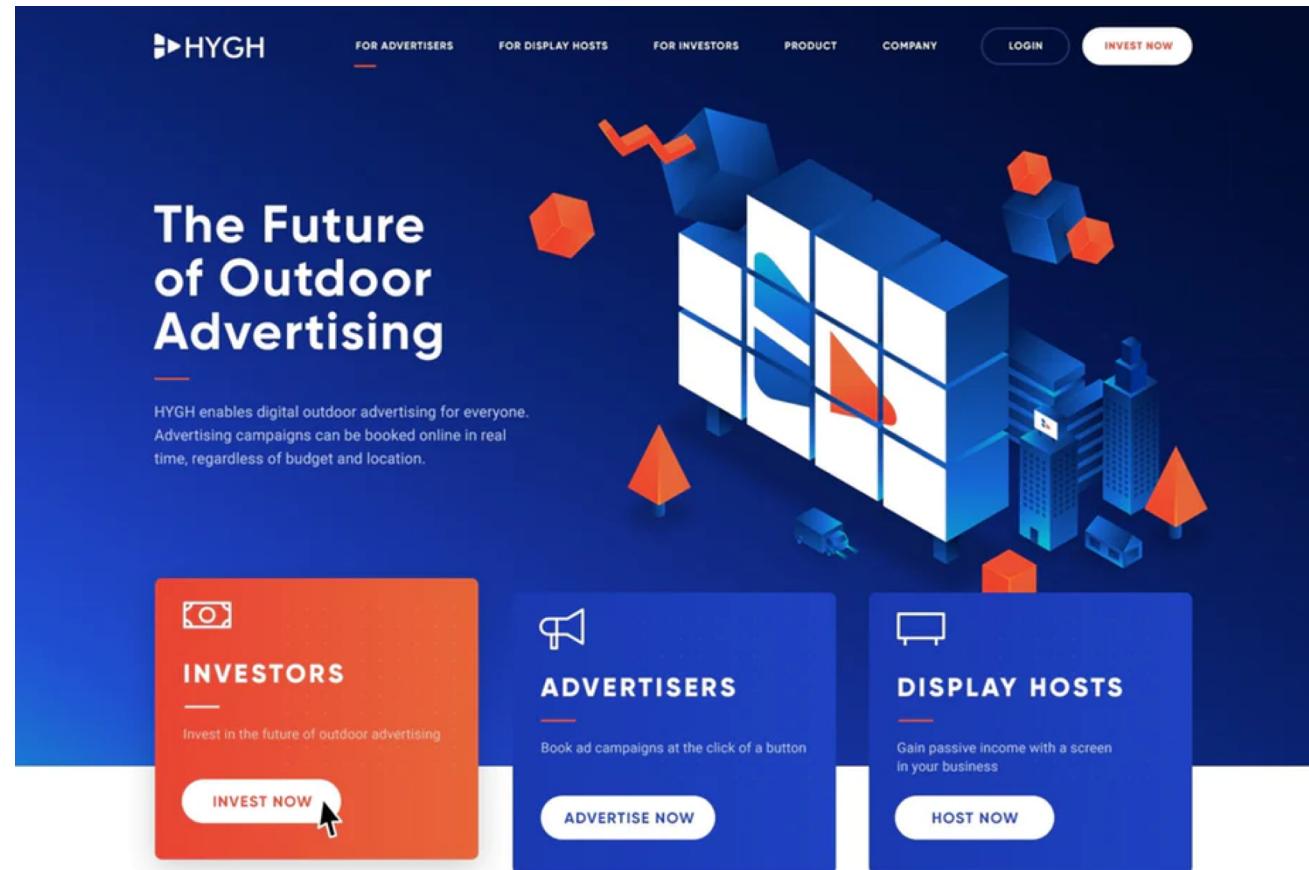


# F.ITM202 – Веб зохиомж



# Cascading Style Sheet

- CSS үргэлжлэл
- CSS3 & Респонсив

# Dropdown menu

```
.dropbtn {  
    background-color: #4CAF50;  
    color: white;  
    padding: 16px;  
    font-size: 16px;  
    border: none;  
    cursor: pointer;  
}
```

```
.dropdown {  
    position: relative;  
    display: inline-block;  
}
```

```
.dropdown-content a {  
    color: black;  
    padding: 12px 16px;  
    text-decoration: none;  
    display: block;  
}
```

```
.dropdown-content {  
    display: none;  
    position: absolute;  
    background-color: #f9f9f9;  
    min-width: 160px;  
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);  
    z-index: 1;  
}
```

```
.dropdown-content a:hover {background-color: #f1f1f1}  
  
.dropdown:hover .dropdown-content {  
    display: block;  
}  
  
.dropdown:hover .dropbtn {  
    background-color: #3e8e41;  
}
```

# Dropdown menu

```
<body>

<h2>Dropdown Menu</h2>
<p>Move the mouse over the button to open the dropdown menu.</p>

<div class="dropdown">
  <button class="dropbtn">Dropdown</button>
  <div class="dropdown-content">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
  </div>
</div>

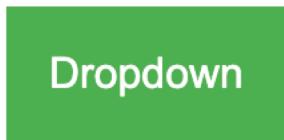
<p><strong>Note:</strong> We use href="#" for test links. In a real web site this would be URLs.</p>

</body>
</html>
```

# Dropdown menu

## Dropdown Menu

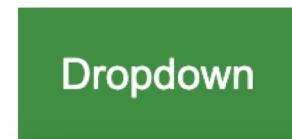
Move the mouse over the button to open the dropdown menu.



**Note:** We use href="#" for test links. In a real

## Dropdown Menu

Move the mouse over the button to open the dropdown menu.



Link 1

Link 2

Link 3

or test links. In a real web site this would be URLs.

# Image gallery

```
div.gallery {  
    border: 1px solid #ccc;  
}  
  
div.gallery:hover {  
    border: 1px solid #777;  
}
```

```
div.gallery img {  
    width: 100%;  
    height: auto;  
}  
  
div.desc {  
    padding: 15px;  
    text-align: center;  
}
```

```
.responsive {  
    padding: 0 6px;  
    float: left;  
    width: 24.99999%;  
}  
  
@media only screen and (max-width: 700px) {  
    .responsive {  
        width: 49.99999%;  
        margin: 6px 0;  
    }  
}  
  
@media only screen and (max-width: 500px) {  
    .responsive {  
        width: 100%;  
    }  
}
```

# Image gallery

```
<body>

<h2>Responsive Image Gallery</h2>

<h4>Resize the browser window to see the effect.</h4>

<div class="responsive">
  <div class="gallery">
    <a target="_blank" href="img_5terre.jpg">
      
    </a>
    <div class="desc">Add a description of the image here</div>
  </div>
</div>

<div class="responsive">
  <div class="gallery">
    <a target="_blank" href="img_forest.jpg">
```

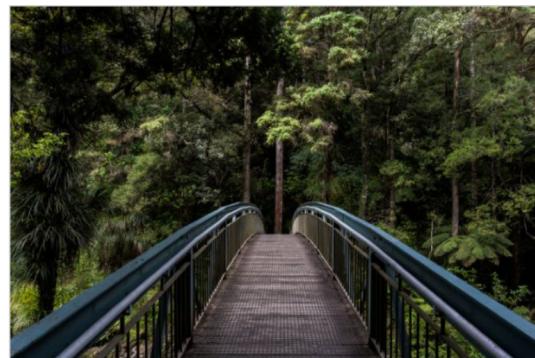
# Image gallery

## Responsive Image Gallery

Resize the browser window to see the effect.



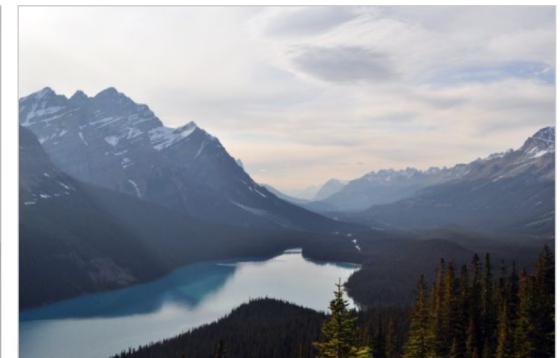
Add a description of the image here



Add a description of the image here



Add a description of the image here



Add a description of the image here

This example uses media queries to re-arrange the images on different screen sizes: for screens larger than 700px wide, it will show four images side by side, for screens smaller than 700px, it will show two images side by side. For screens smaller than 500px, the images will stack vertically (100%).

You will learn more about media queries and responsive web design later in our CSS Tutorial.

# .clearfix::after{ }

```
<style>
div {
    border: 3px solid #4CAF50;
    padding: 5px;
}

.img1 {
    float: right;
}

.img2 {
    float: right;
}

.clearfix::after {
    content: "";
    clear: both;
    display: table;
}
</style>
```

```
<div>
    
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet...
</div>
```

```
<h2 style="clear:right">With New Modern Clearfix</h2>
<p>Add the clearfix hack to the containing element, to fix this problem:</p>
```

```
<div class="clearfix">
    
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet...
</div>
```

## Without Clearfix

This image is floated to the right. It is also taller than the element containing it, so it overflows outside of its container:

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet...



## With New Modern Clearfix

Add the clearfix hack to the containing element, to fix this problem:

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet...



# Attribute selector

```
a[target] {  
    background-color: yellow;  
}
```

```
<a href="http://www.disney.com" target="_blank">disney.com</a>
```

! disney.com !

# Attribute selector

```
a[target="_blank"] {  
    background-color: yellow;  
}
```

```
<a href="https://www.w3schools.com">w3schools.com</a>  
<a href="http://www.disney.com" target="_blank">disney.com</a>  
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>
```

w3schools.com disney.com wikipedia.org

# Attribute selector

```
[title~="flower"] {  
    border: 5px solid yellow;  
}
```

```
  
  

```

All images with the title attribute containing the word "flower" get a yellow border.



# Attribute selector

```
[class|="top"] {  
    background: yellow;  
}
```

```
<h1 class="top-header">Welcome</h1>  
<p class="top-text">Hello world!</p>  
<p class="topcontent">Are you learning CSS?</p>
```

Welcome

Hello world!

Are you learning CSS?

# Attribute selector

```
[class^="top"] {  
    background: yellow;  
}
```

```
<h1 class="top-header">Welcome</h1>  
<p class="top-text">Hello world!</p>  
<p class="topcontent">Are you learning CSS?</p>
```

Welcome

Hello world!

Are you learning CSS?

# Attribute selector

```
[class$="test"] {  
    background: yellow;  
}
```

```
<div class="first_test">The first div element.</div>  
<div class="second">The second div element.</div>  
<div class="my-test">The third div element.</div>  
<p class="mytest">This is some text in a paragraph.</p>
```

The first div element.

The second div element.

The third div element.

This is some text in a paragraph.

# Attribute selector

```
[class*="te"] {  
    background: yellow;  
}
```

```
<div class="first_test">The first div element.</div>  
<div class="second">The second div element.</div>  
<div class="my-test">The third div element.</div>  
<p class="mytest">This is some text in a paragraph.</p>
```

The first div element.

The second div element.

The third div element.

This is some text in a paragraph.

# Counters

```
body {  
  counter-reset: section;  
}  
  
h1 {  
  counter-reset: subsection;  
}  
  
h1::before {  
  counter-increment: section;  
  content: "Section " counter(section) ". ";  
}  
  
h2::before {  
  counter-increment: subsection;  
  content: counter(section) "." counter(subsection) " ";  
}
```

<h1>HTML/CSS Tutorials</h1>  
<h2>HTML</h2>  
<h2>CSS</h2>  
<h2>Bootstrap</h2>  
<h2>W3.CSS</h2>

<h1>Scripting Tutorials</h1>  
<h2>JavaScript</h2>  
<h2>jQuery</h2>  
<h2>React</h2>

<h1>Programming Tutorials</h1>  
<h2>Python</h2>  
<h2>Java</h2>  
<h2>C++</h2>



## Section 1. HTML/CSS Tutorials

1.1 HTML

1.2 CSS

1.3 Bootstrap

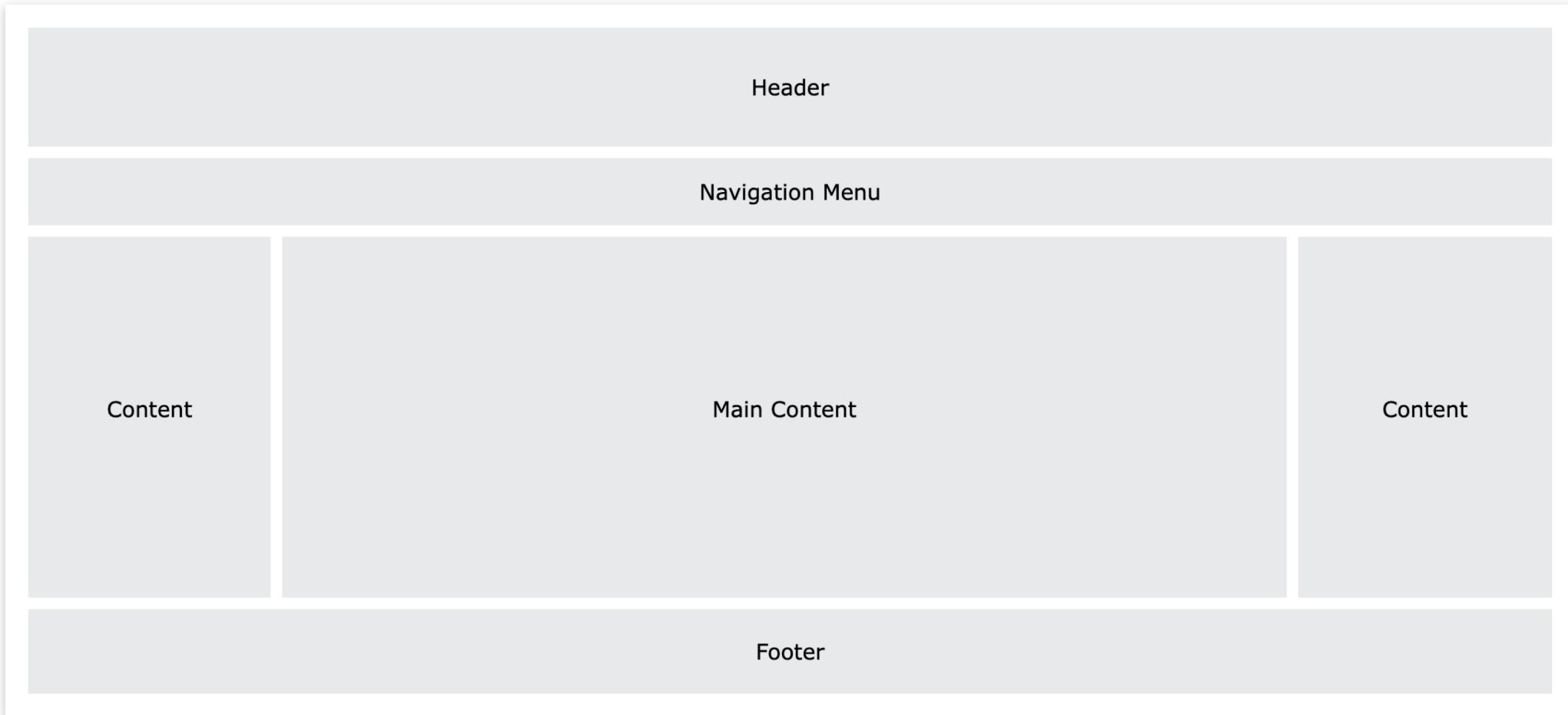
1.4 W3.CSS

## Section 2. Scripting Tutorials

2.1 JavaScript

2.2 jQuery

# Website Layout



# Math function

```
#div1 {  
  position: absolute;  
  left: 50px;  
  width: calc(100% - 100px);  
  border: 1px solid black;  
  background-color: yellow;  
  padding: 5px;  
}
```

`<p>`Create a div that stretches across  
and the edges of the window:`</p>`

```
<div id="div1">Some text...</div>  
</body>  
</html>
```

## The calc() Function

Create a div that stretches across the window, with a 50px gap between both sides of the div and the edges of the window:

Some text...

# Shadow effect

- text-shadow

**Text shadow effect!**

- box-shadow

A <div> element with a box-shadow

**Text shadow effect!**

**Text shadow effect!**

**Text shadow effect!**

A <div> element with a blurred, lightblue box-shadow, with a spread radius of 12px

# CSS3 Text Properties

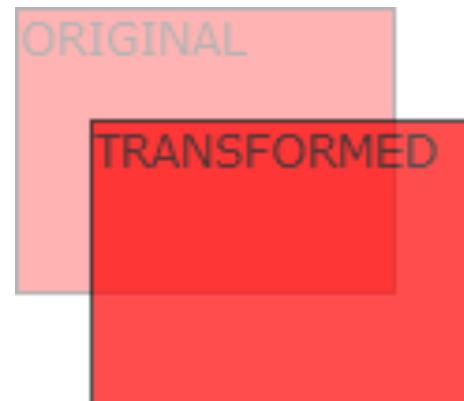
Property	Description
<u>hanging-punctuation</u>	Specifies whether a punctuation character may be placed outside the line box
<u>punctuation-trim</u>	Specifies whether a punctuation character should be trimmed
text-align-last	Describes how the last line of a block or a line right before a forced line break is aligned when text-align is "justify"
<u>text-emphasis</u>	Applies emphasis marks, and the foreground color of the emphasis marks, to the element's text
<u>text-justify</u>	Specifies the justification method used when text-align is "justify"
<u>text-outline</u>	Specifies a text outline
<u>text-overflow</u>	Specifies what should happen when text overflows the containing element
<u>text-shadow</u>	Adds shadow to text
<u>text-wrap</u>	Specifies line breaking rules for text
<u>word-break</u>	Specifies line breaking rules for non-CJK scripts
<u>word-wrap</u>	Allows long, unbreakable words to be broken and wrap to the next line

# Хувиргалт

- transform
  - translate()
  - rotate()
  - scale()
  - skew()
  - matrix()

# translate()

```
div {  
    -ms-transform: translate(50px,100px);  
    /* IE 9 */  
    -webkit-transform: translate(50px,100px);  
    /* Chrome, Safari, Opera */  
    transform: translate(50px,100px);  
}
```



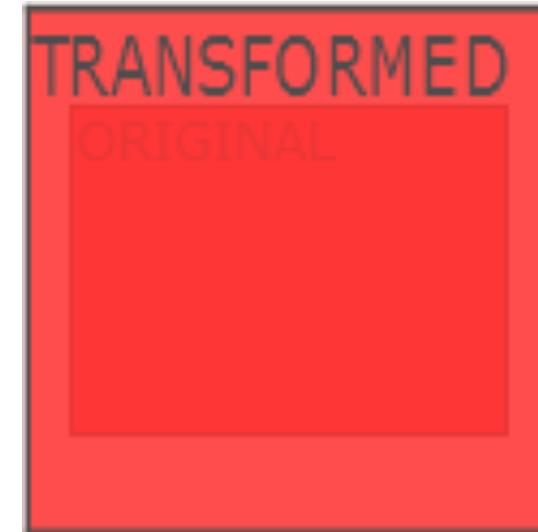
# rotate()

```
div {  
    -ms-transform: rotate(30deg);  
    /* IE 9 */  
    -webkit-transform: rotate(30deg);  
    /* Chrome, Safari, Opera */  
    transform: rotate(30deg);  
}
```



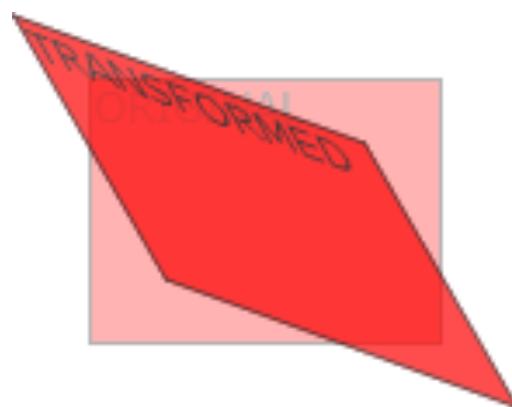
# scale()

```
div {  
    -ms-transform: scale(2,4);  
    /* IE 9 */  
    -webkit-transform: scale(2,4);  
    /* Chrome, Safari, Opera */  
    transform: scale(2,4);  
}
```



# skew()

```
div {  
    -ms-transform: skew(30deg,20deg);  
    /* IE 9 */  
    -webkit-transform: skew(30deg,20deg);  
    /* Chrome, Safari, Opera */  
    transform: skew(30deg,20deg);  
}
```



# matrix()

```
div {  
    -ms-transform: matrix(0.866,0.5,-0.5,0.866,0,0);  
    /* IE 9 */  
    -webkit-transform: matrix(0.866,0.5,-0.5,0.866,0,0);  
    /* Chrome, Safari, Opera */  
    transform: matrix(0.866,0.5,-0.5,0.866,0,0);  
}
```



# 2D Transform Methods

- `matrix(n,n,n,n,n,n)` Defines a 2D transformation, using a matrix of six values
- `translate(x,y)` Defines a 2D translation, moving the element along the X- and the Y-axis
- `translateX(n)` Defines a 2D translation, moving the element along the X-axis
- `translateY(n)` Defines a 2D translation, moving the element along the Y-axis
- `scale(x,y)` Defines a 2D scale transformation, changing the elements width and height
- `scaleX(n)` Defines a 2D scale transformation, changing the element's width
- `scaleY(n)` Defines a 2D scale transformation, changing the element's height
- `rotate(angle)` Defines a 2D rotation, the angle is specified in the parameter
- `skew(x-angle,y-angle)` Defines a 2D skew transformation along the X- and the Y-axis
- `skewX(angle)` Defines a 2D skew transformation along the X-axis
- `skewY(angle)` Defines a 2D skew transformation along the Y-axis

# Хөдөлгөөн

- @keyframes
- animation

# @keyframes

- A name of our choosing (tutsFade in this case).
- Stages: 0%-100%; from (equal to 0%) and to (equal to 100%).
- CSS styles: the style that you would like to apply for each stage.

# ЖИШЭЭ

```
@keyframes tutsFade {  
    0% {  
        opacity: 1;  
    }  
    100% {  
        opacity: 0;  
    }  
}
```

```
@keyframes examp {  
    0% {background-color: red;}  
    25% {background-color: yellow;}  
    50% {background-color: blue;}  
    100% {background-color: green;}  
}
```

```
@keyframes example {  
    from {background-color: red;}  
    to {background-color: yellow;}  
}
```

# Animation

- animation-name: @keyframes name (remember we chose tutsFade).
- animation-duration: the timeframe length, the total duration of the animation from start to the end.
- animation-timing-function: sets the animation speed ( linear | ease | ease-in | ease-out | ease-in-out | cubic-bezier ).
- animation-delay: the delay before our animation will start.
- animation-iteration-count: how many times it will iterate through animation.
- animation-direction: gives you the ability to change the loop direction, from start to end ,or from end to start, or both.
- animation-fill-mode: specifies which styles will be applied to the element when our animation is finished ( none | forwards | backwards | both )

# Animation

```
.element {  
    animation-name: tutsFade;  
    animation-duration: 4s;  
    animation-delay: 1s;  
    animation-iteration-count: infinite;  
    animation-timing-function: linear;  
    animation-direction: alternate;  
}
```

# Multiple Animations

```
.element {  
    animation: tutsFade 4s 1s infinite linear alternate,  
    tutsRotate 4s 1s infinite linear alternate;  
}  
  
@keyframes tutsFade {  
    to {  
        opacity: 0;  
    }  
}  
  
@keyframes tutsRotate {  
    to {  
        transform: rotate(180deg);  
    }  
}
```

# Transitions

- Transition
- Transition-delay
- Transition-duration
- Transition-property
- Transition-timing-function

```
div {  
    -webkit-transition: width 2s; /* For Safari 3.1 to 6.0 */  
    transition: width 2s;  
}  
div:hover {  
    width: 300px;  
}
```

```
div {  
    -webkit-transition: width 2s, height 2s,  
    -webkit-transform 2s; /* For Safari 3.1 to 6.0 */  
    transition: width 2s, height 2s, transform 2s;  
}
```

```
div {  
/* For Safari 3.1 to 6.0 */  
-webkit-transition-property: width;  
-webkit-transition-duration: 1s;  
-webkit-transition-timing-function: linear;  
-webkit-transition-delay: 2s;  
/* Standard syntax */  
transition-property: width;  
transition-duration: 1s;  
transition-timing-function: linear;  
transition-delay: 2s;  
}  
div {  
-webkit-transition: width 1s linear 2s; /* For Safari 3.1 to 6.0 */  
transition: width 1s linear 2s;  
}
```

# Багана

- Column-count
- Column-gap
- Column-rule

```
div {  
    -webkit-column-count: 3; /* Chrome, Safari, Opera */  
    -moz-column-count: 3; /* Firefox */  
    column-count: 3;  
}
```

```
div {  
    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */  
    -moz-column-gap: 40px; /* Firefox */  
    column-gap: 40px;  
}
```

```
div {  
    -webkit-column-rule: 3px outset #ff00ff; /* Chrome, Safari, Opera */  
    -moz-column-rule: 3px outset #ff00ff; /* Firefox */  
    column-rule: 3px outset #ff00ff;  
}
```

# Респонсив загвар

- Мобайл төхөөрөмжийн хэрэглээ эрс нэмэгдэж байна
- Гар утас, таблет, том жижиг компьютер тус бүрд зориулсан вэбүүд хэрэгтэй

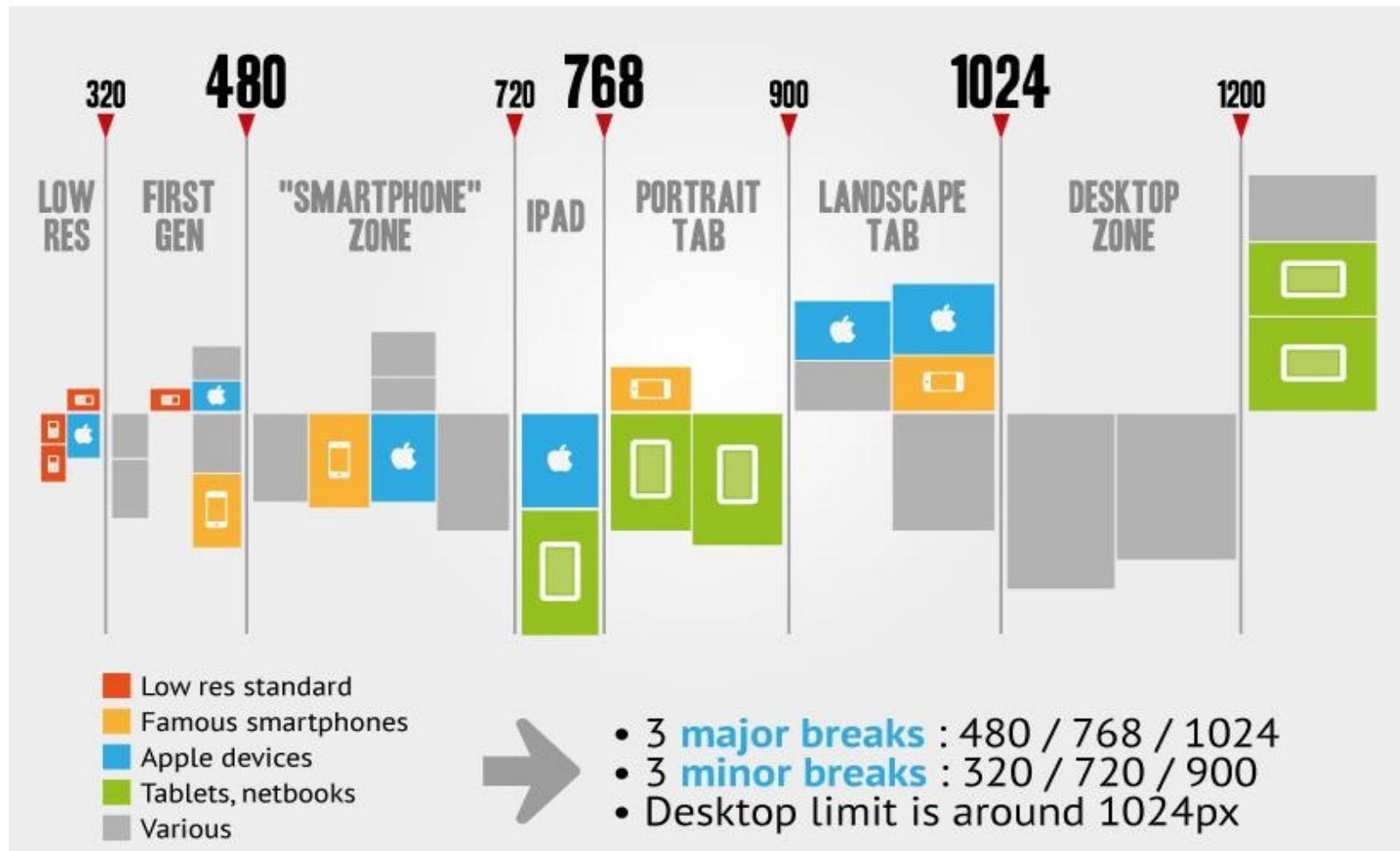


# Респонсив загвар

- Респонсив загвар ашигласнаар төхөөрөмж бүрт зориулж өөр өөр вэб хийх шаардлагагүй
- Төхөөрөмжийн дэлгэцийн хэмжээнээс шалтгаалж харагдах хэлбэрээ өөрчилнө



# Дэлгэцийн нягтрал



# Өөрчлөлтийн цэг тодорхойлох

	<b>&lt;480</b>
	<b>&lt;768</b>
	<b>&gt;768</b>
	<b>&lt;320</b>
	<b>768-1024</b>
	<b>&gt;1024</b>

## Үндсэн

- Эхэн үеийн ухаалаг утаснуудын босоо дэлгэцээр үзэх үед **<480px** нөхцлийг ашиглана
- Орчин үеийн ухаалаг утас болон таблетын босоо дэлгэцээр үзэх үед **<768px** нөхцлийг ашиглана
- Бусад том таблет, компьютерийн дэлгэцээр үзэх үед **>768px** нөхцлийг ашиглана

## Нэмэлт

- Бага нягтралтай дэлгэцээр үзэх үед **<320px** нөхцлийг ашиглана
- Том таблет болон таблетын хөндлөн дэлгэцээр үзэх үед **>768px AND <1024px** нөхцлийг ашиглана
- Өндөр нягтралтай компьютерийн дэлгэцээр үзэх үед **>1024px** нөхцлийг ашиглана

# Зураг

- Бага дэлгэц дээрх зургийг жижиг зургаар солих
- `img{ width: 100%; }`
- `img { max-width: 100%; }`

# MAX & MIN заавал тодорхойлох

**Fluid, but not under 768  
and not over 1024:**

```
min-width: 768px;  
max-width: 1024px;
```

**At least 350 high:**

```
min-height: 350px;
```

**#main not under 1000 wide  
when viewed on a desktop:**

```
@media screen and (min-width:  
1024px) {  
  #main { min-width: 1000px; }  
}
```

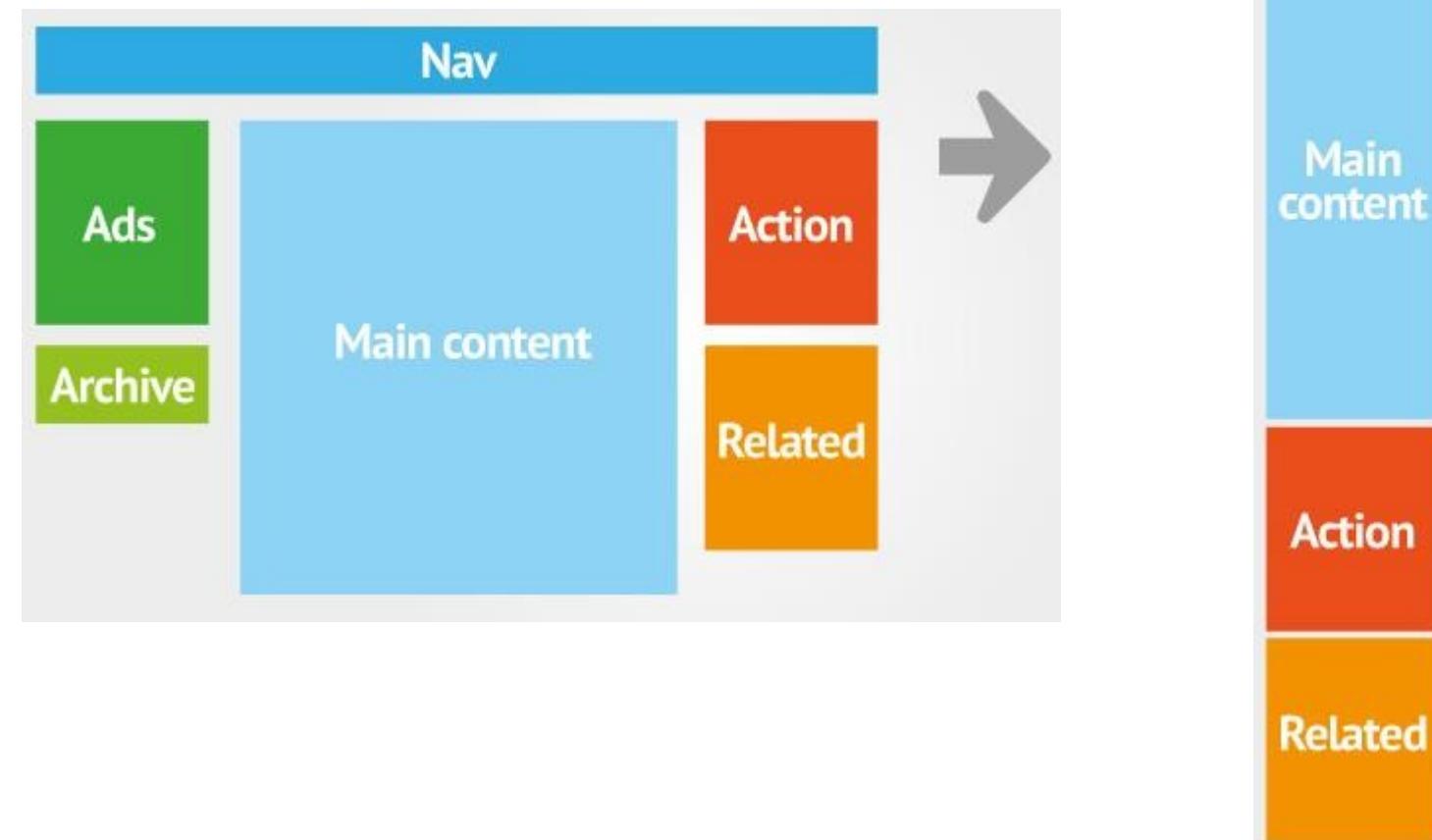
# Уян хатан хэмжээс

- Агуулагч тагийн өргөнийг тогтмолоор зарлаж, дотор тагуудын өргөнийг хувиар тодорхойлох
- Emphemeral хэмжээс ашиглах

With master *font-size* = 16px;

- width: 46.25em                      -> 46.25x16    = 740px
- border-width: 0.063em              -> 0.063x16    = 1px
- font-size: 1.125em;                -> 1.125x16    = 18px
- padding: 0.325em                  -> 0.325x16    = 5px

# Мобайл төхөөрөмжөөс үзэхэд нэг баганатай болгох

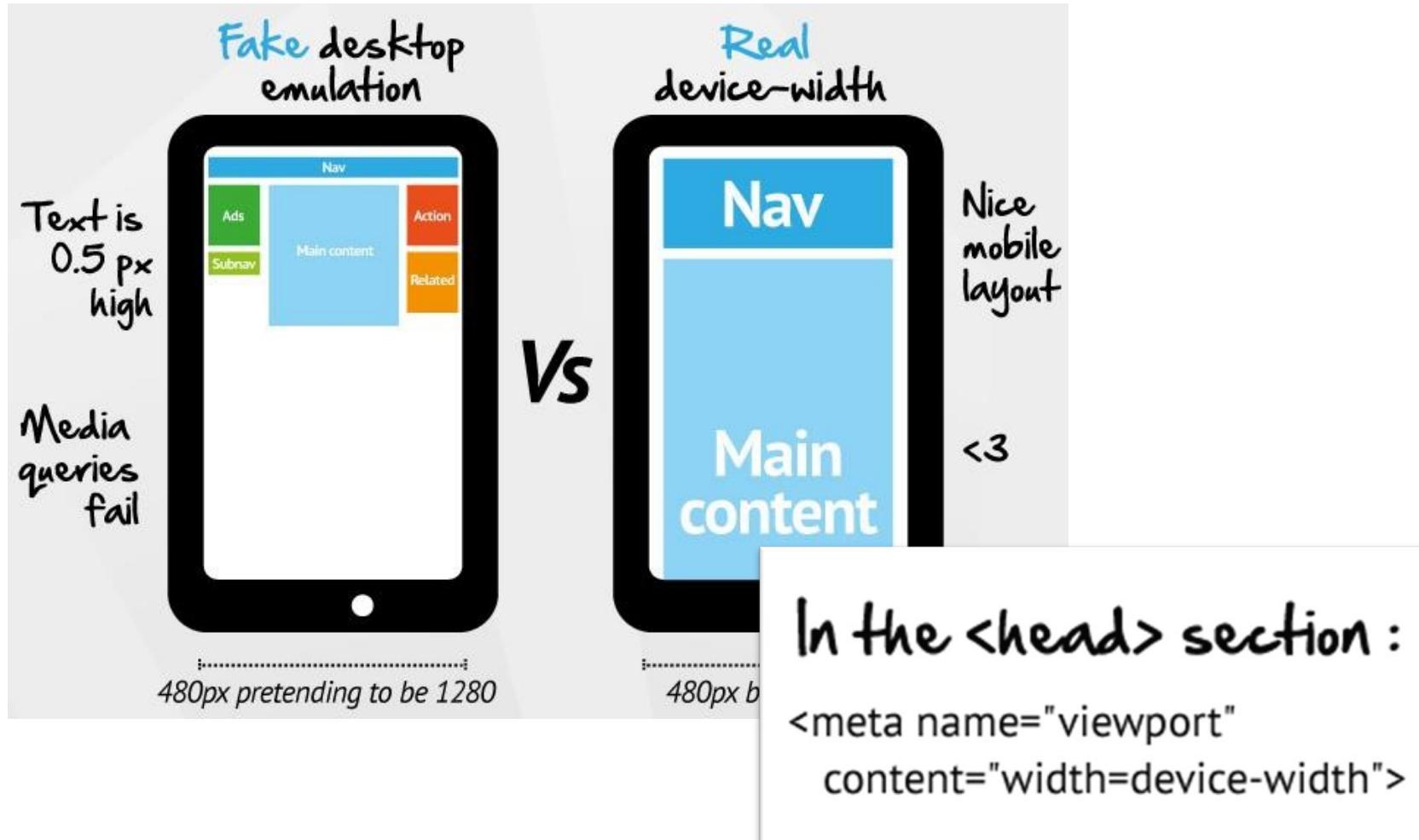




## Чухал биш хэсгийг нуух

```
@media screen and (max-width: 300px) {  
    .not_mobile { display: none; }  
}
```

# Viewport тодорхойлж өгөх



Анхаарал тавьсанд  
баярлалаа