

# #TasteOfCode

Intro to programming

:{) Codaisseur

MIRABEAU \*

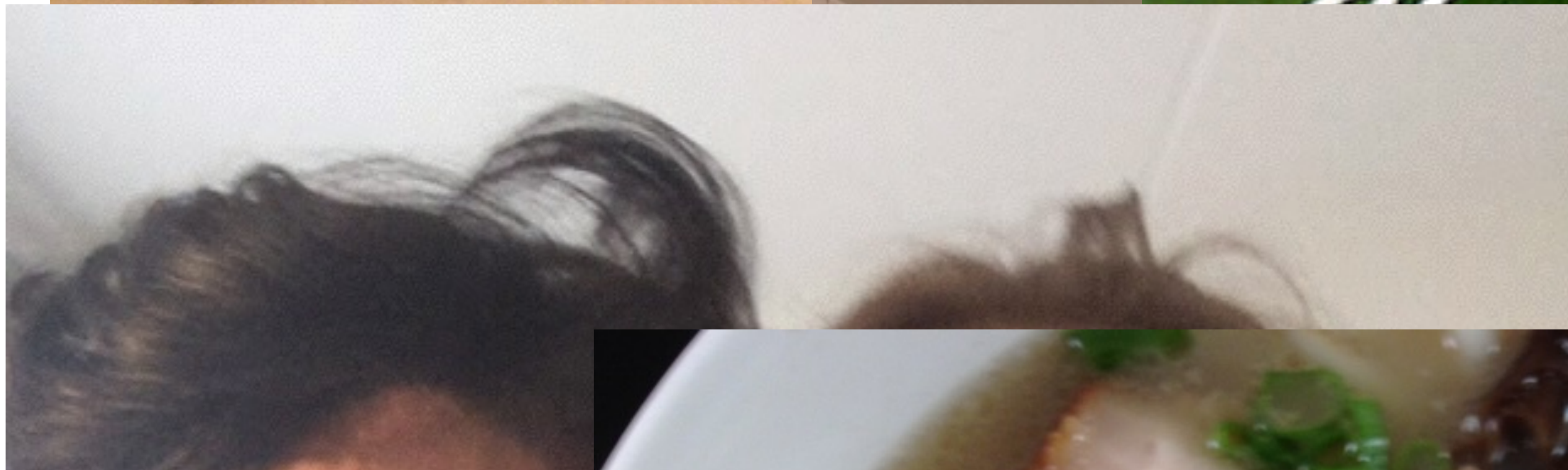
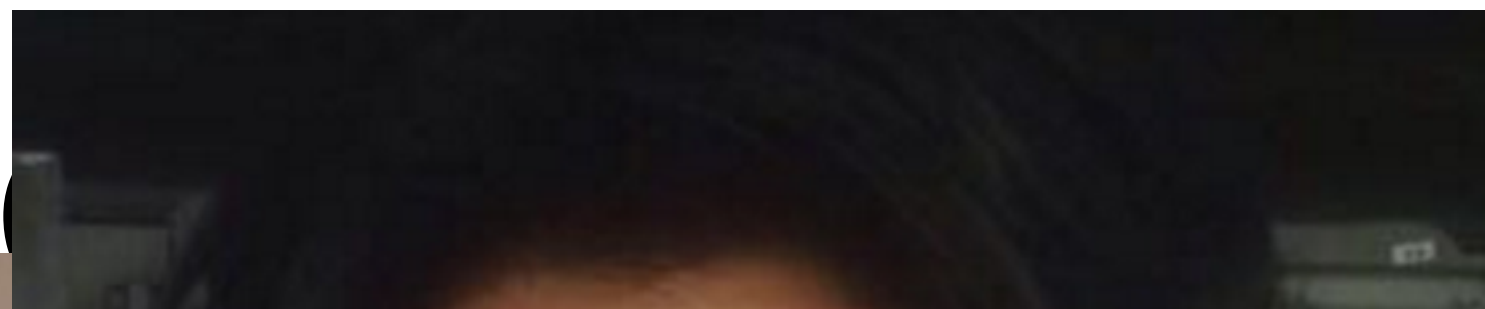
le comptoir  
de l'innovation

J.P.Morgan

xerox 



Rory Sid



Championship  
**MANAGER**





# Wat hebben we nodig?

- Code schrijven: [codepen.io](https://codepen.io)
- Publiceren (later): [bitballoon.com](https://bitballoon.com)

# HTML



**:{) Codaisseur**

# Structure

schedule.html

- `<h1>` Heading - level 1
- `<p>` Paragraph
- `<ul>` Unordered List
- `<li>` List item

```
<h1>Agenda</h1>
```

```
<p>On the menu today:</p>
```

```
<ul>
```

```
  <li>Structure with HTML</li>
```

```
  <li>Styling with CSS</li>
```

```
  <li>Happiness with Lunch</li>
```

```
</ul>
```

**:{) Codaisseur**

# Render

schedule.html

`<h1>Agenda</h1>`

`<p>On the menu today:</p>`

`<ul>`

`<li>Structure with HTML</li>`

`<li>Styling with CSS</li>`

`<li>Happiness with Lunch</li>`

`</ul>`



## Agenda

On the menu today:

- Structure with HTML
- Styling with CSS
- Happiness with Lunch

`:{)` Codaisseur

# TAGS open...

```
<h1>Agenda
```

Opening tag - starts heading

# TAGS open and close

```
<h1>Agenda</h1>
```

**Opening** tag - starts heading

**Closing** tag - stops heading



# Applying TAGS

index.html

Taste of Code We will build a game.

## Browser result



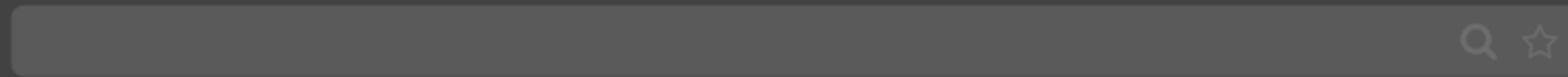
Taste of Code We will build a game.

# Open a TAG...

index.html

```
<h1>Taste of Code We will build a game.
```

## Browser result



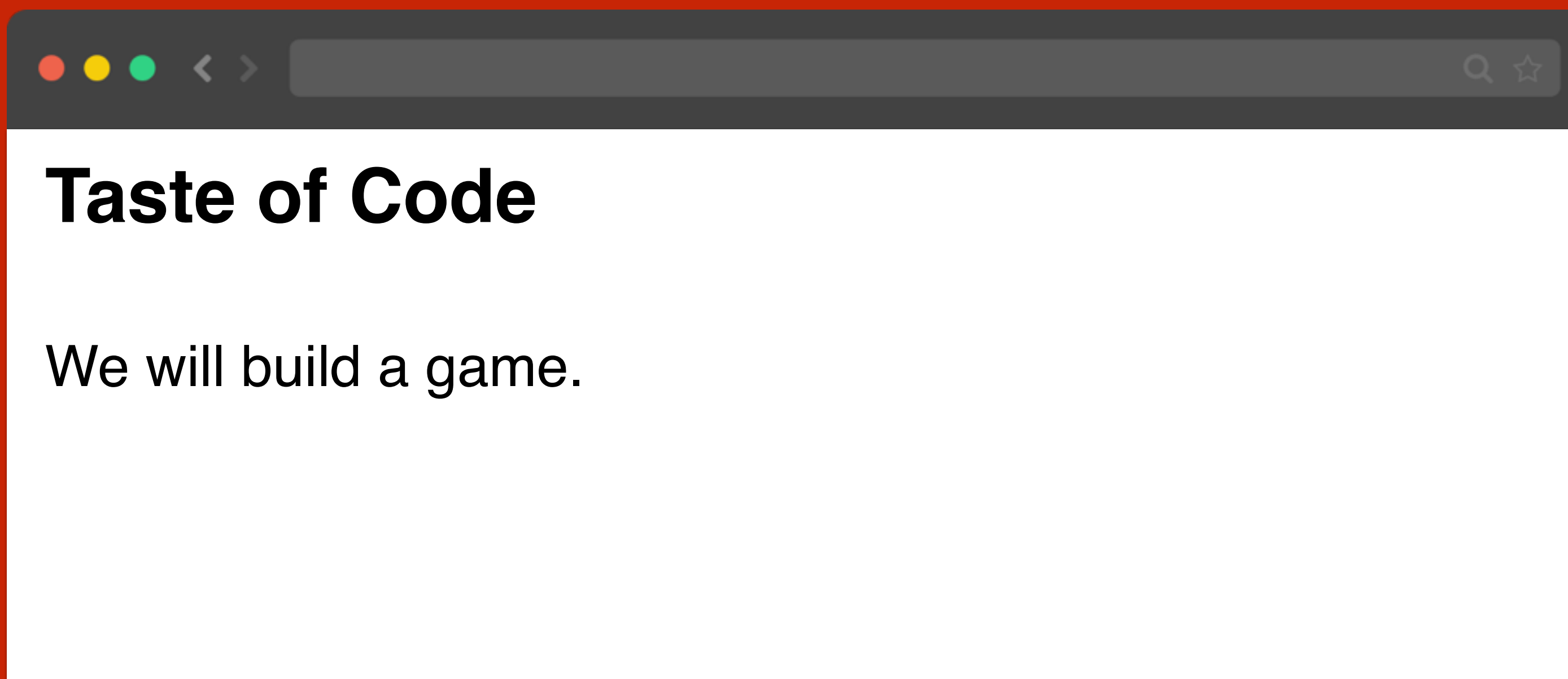
**Taste of Code We will build a game.**

# ... and close a TAG

index.html

```
<h1>Taste of Code</h1>  
We will build a game.
```

## Browser result

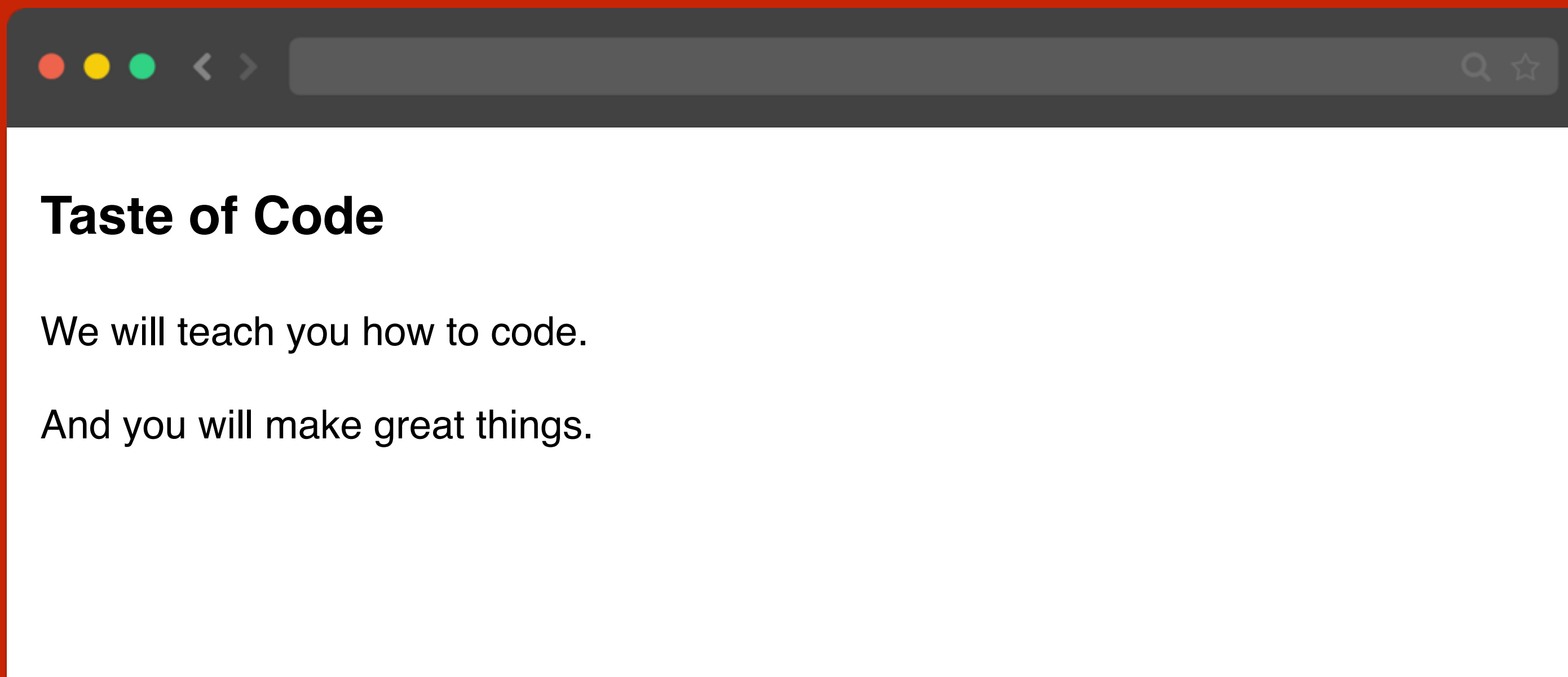


# Another example: Paragraphs

index.html

```
<h1>Taste of Code</h1>
<p>We will teach you how to code.</p>
<p>And you will make great things.</p>
```

## Browser result





```
<!DOCTYPE html>
```

```
<html>
```

```
  <head></head>
```

```
  <body></body>
```

```
</html>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head></head>
```

```
  <body></body>
```

```
</html>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head></head>
```

```
  <body></body>
```

```
</html>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head></head>
```

```
  <body></body>
```

```
</html>
```



10 min

## EXERCISE 1:

**A small HTML document  
with H1 and P**

[codepen.io](https://codepen.io)

# css



**:{) Codaisseur**

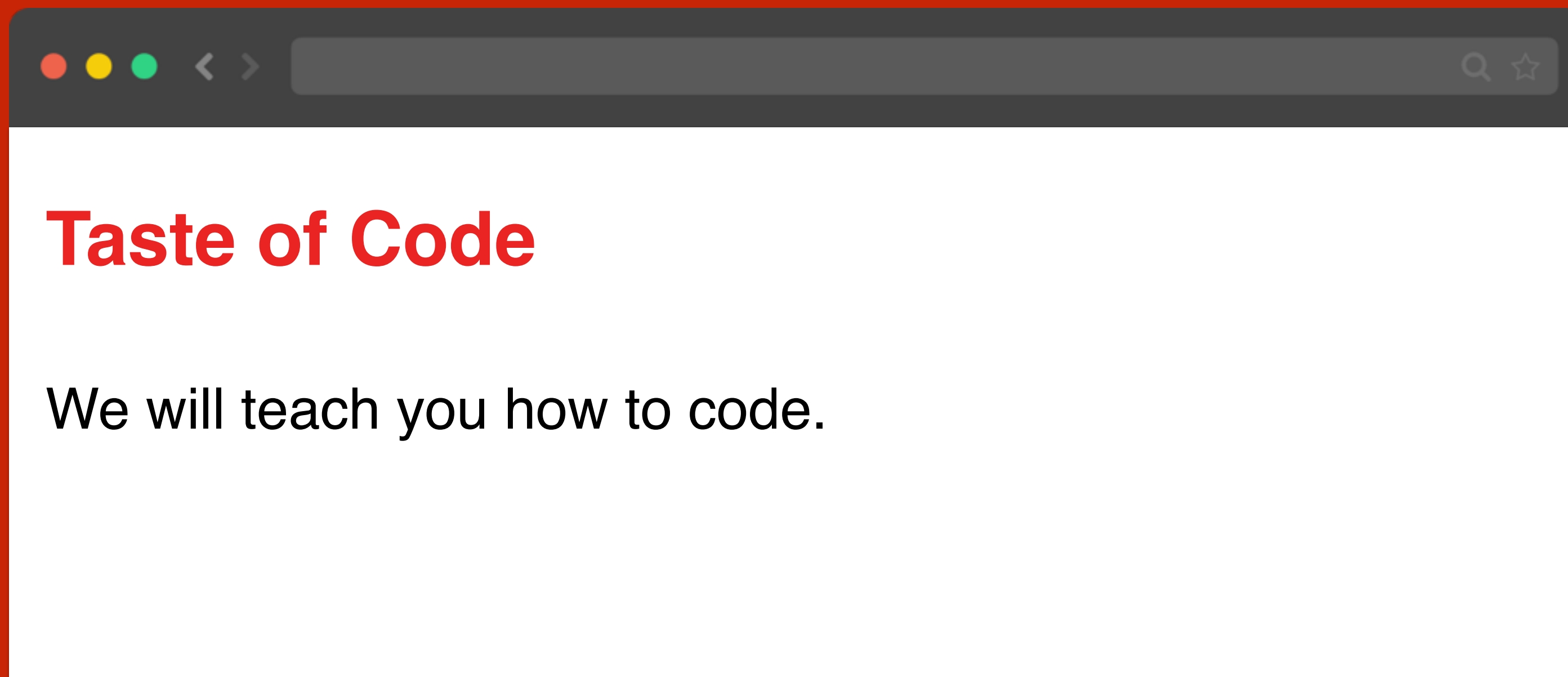
# CSS describes how HTML looks

- color and background
- alignment
- font
- height and width
- padding and margin

# Define color

```
h1 {  
  color: red;  
}
```

## Browser result





```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <style>
```

```
      h1 {
```

```
        color: red;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body></body>
```

```
</html>
```

5 min

## EXERCISE 2:

**Change the background-color  
of  
the HTML page to green**

**.class { }**

# Define color on a class

index.html

```
<h1 class="warning">Taste of Code</h1>  
<p class="warning">We will build a game.</p>
```

```
.warning {  
  color: red;  
}
```

Browser result



**Taste of Code**

We will build a game.



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <style>
```

```
      .warning {
```

```
        color: red;
```

```
      }
```

```
    </style>
```

```
  </head>
```

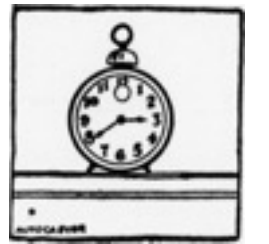
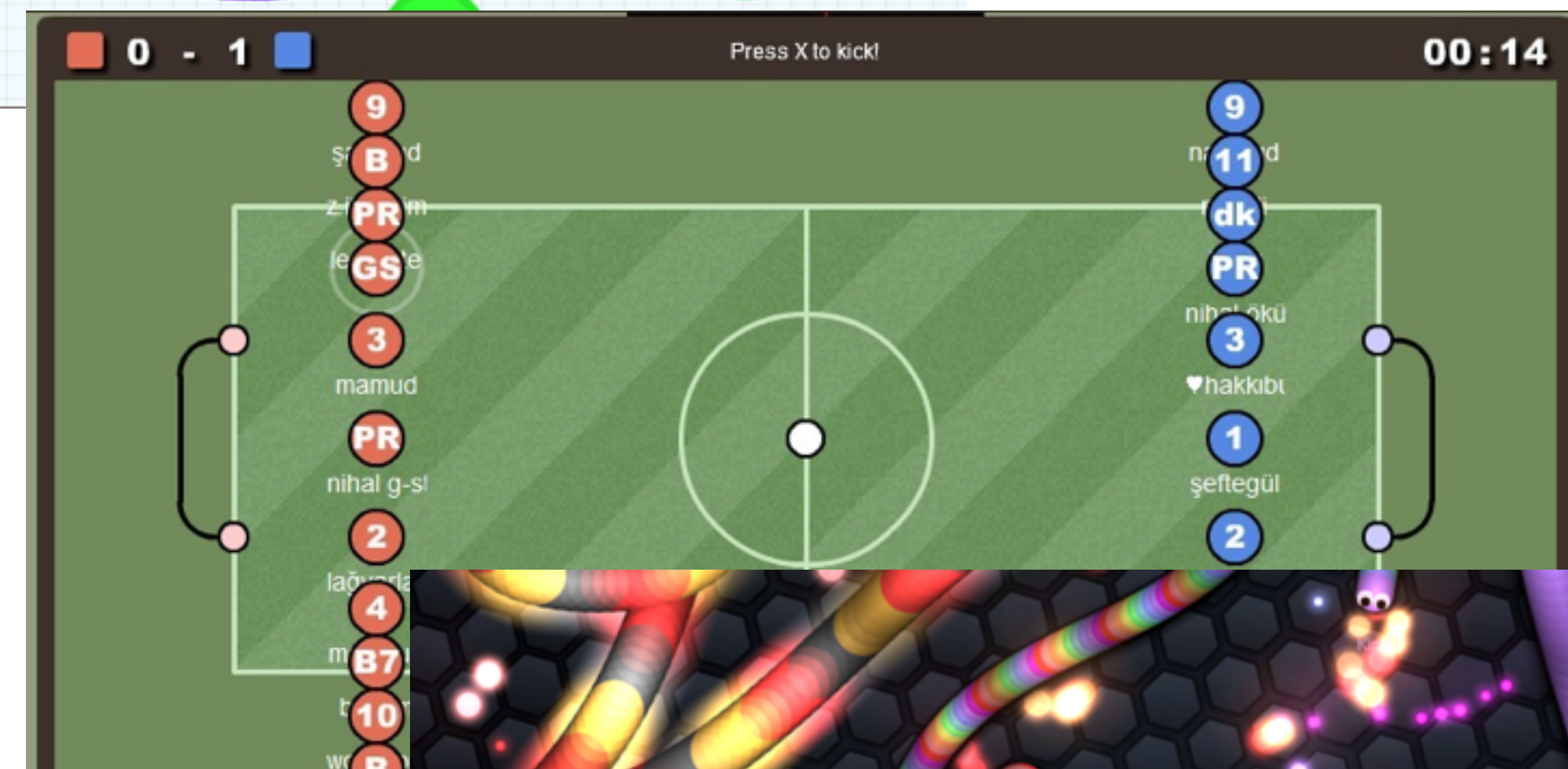
```
  <body></body>
```

```
</html>
```

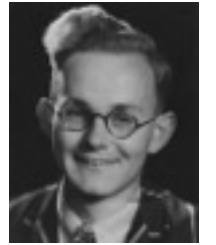


# Pauze!

- [haxball.com](http://haxball.com)
- [agar.io](http://agar.io)
- [slither.io](http://slither.io)
- [curvefever.io](http://curvefever.io)
- Telegramautootje
- kattenfilmpjes
- ?



7



31



10 min

## EXERCISE 3:

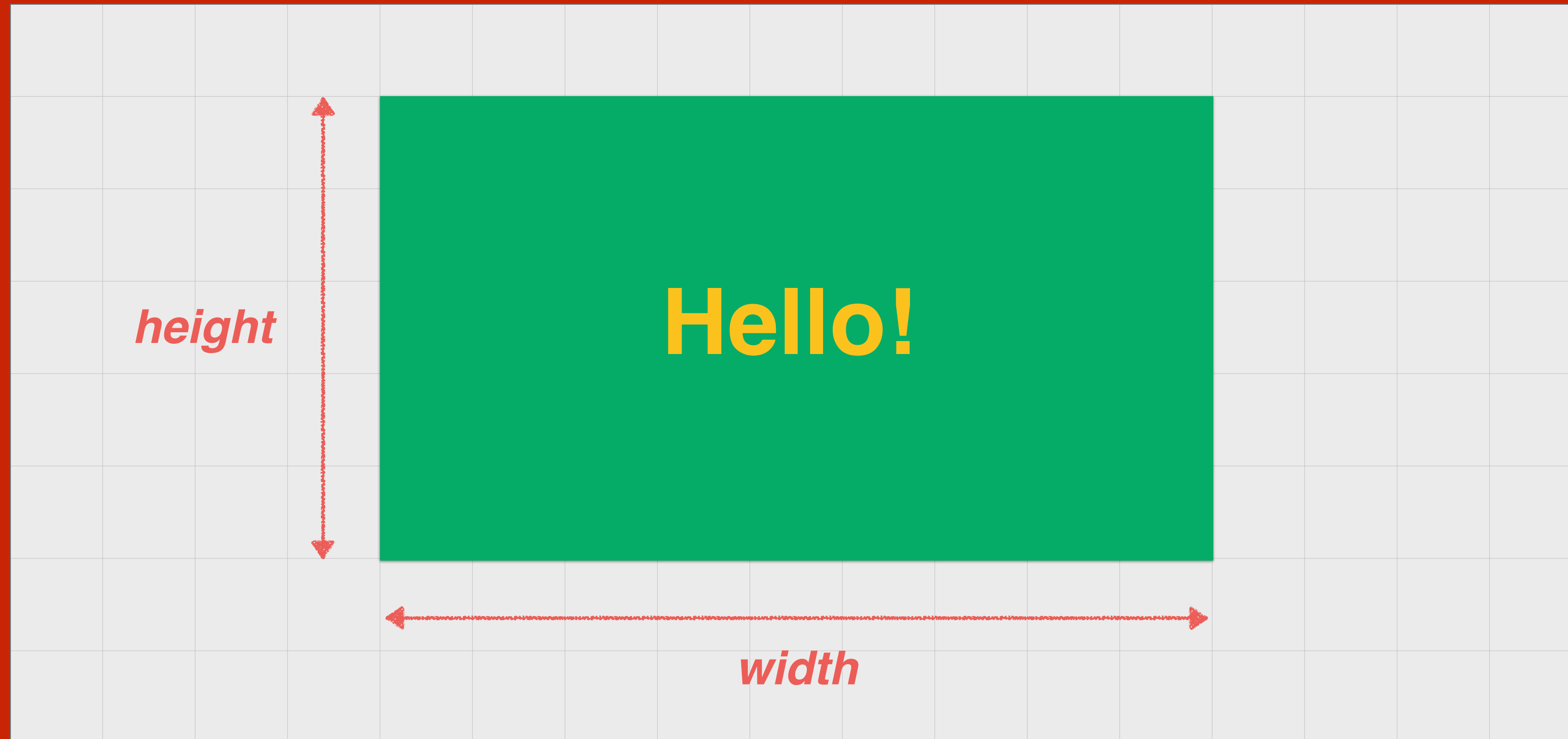
**Apply the warning  
class to <H1>**

PIXELS

# A Screen Consists of Pixels

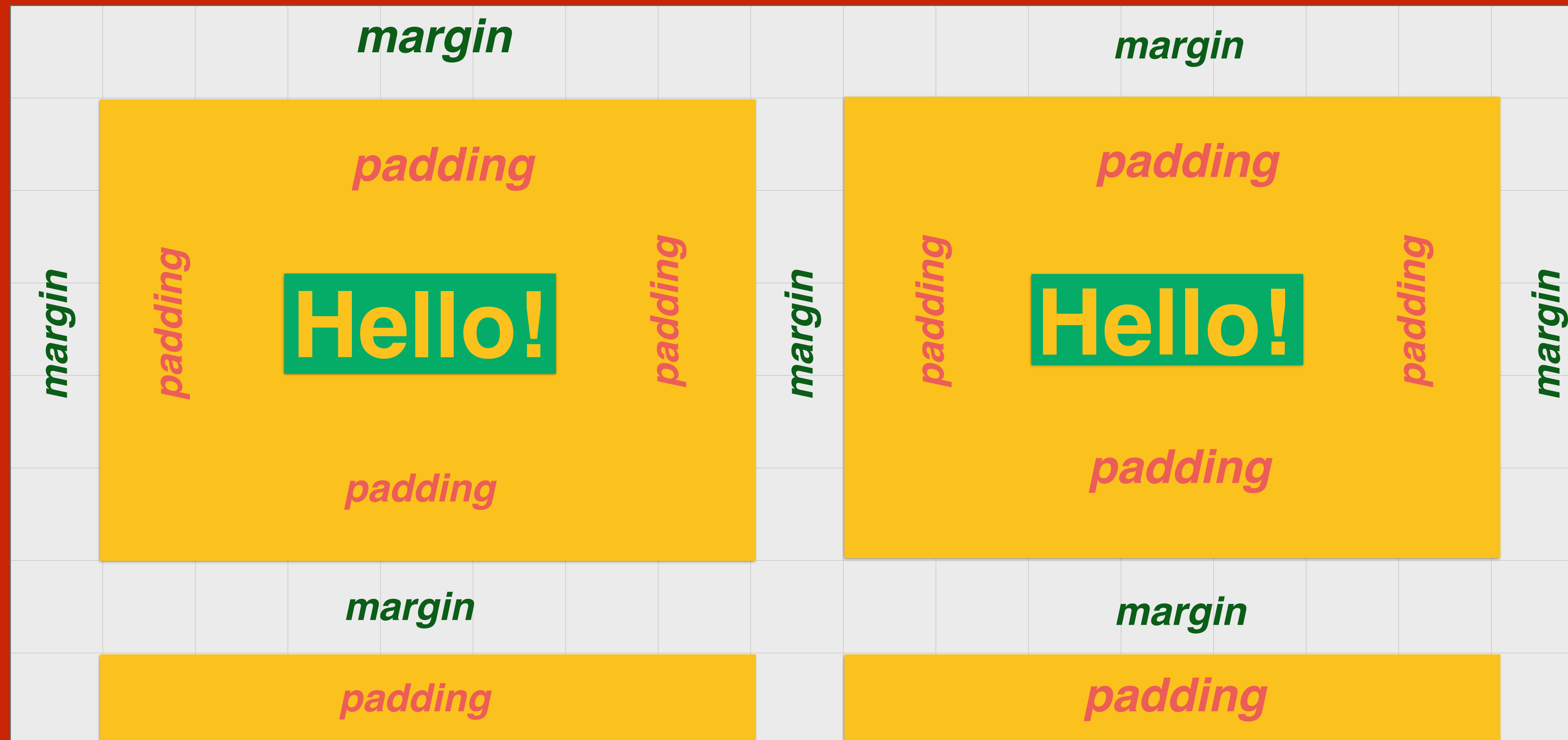


# A Screen Consists of Pixels





# Padding and Margin



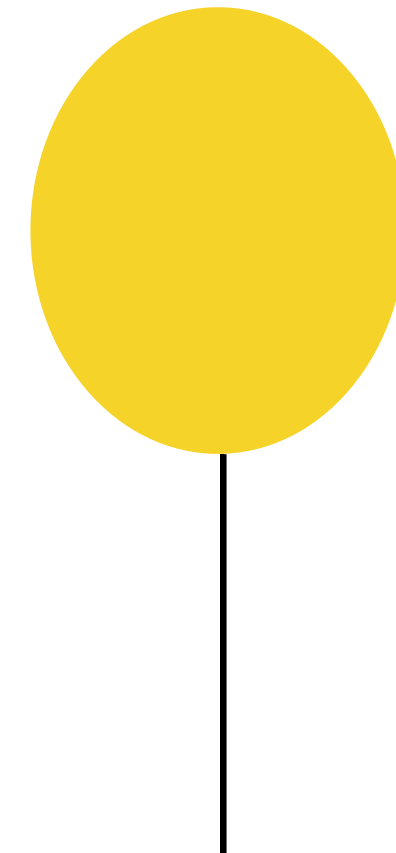
# Draw a balloon!

index.html

```
<div class="balloon">
  <div class="bubble">
  </div>
  <div class="string">
  </div>
</div>
```

style.css

```
.bubble {
  width: 180px;
  height: 200px;
  background: yellow;
  border-radius: 50%;
}
.string {
  height: 100px;
  width: 1px;
  background: black;
  margin-left: 90px;
}
```



10 min

## EXERCISE 4:

Use classes to  
style a balloon

# JS



**:{) Codaisseur**

# HTML CSS JS



Content &  
Structure



Presentation



Behaviour

**PROGRAMMIN**

**G**



# Talk to the browser

*Javascript's possibilities  
are defined by the browser*

**:{) Codaisseur**

**window.alert();**



**object**



**method**

**:{) Codaisseur**

**Object**

# Method

```
function() {}
```

5 min

## EXERCISE 5:

Use Javascript to  
`document.write("This  
is \a program");`

5 min

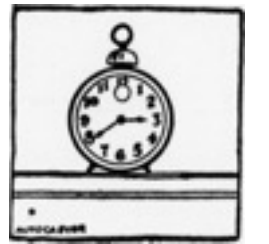
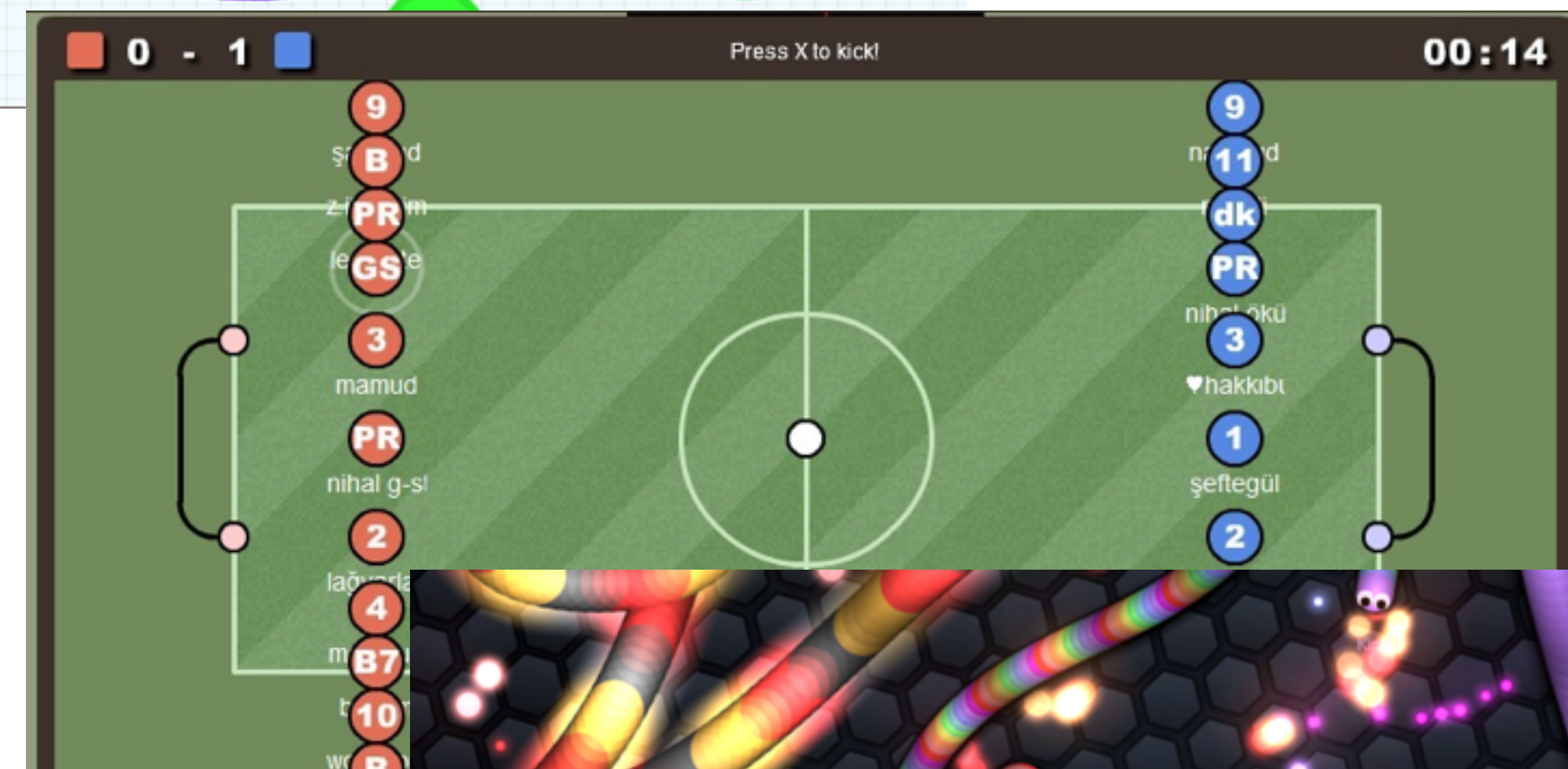
## EXERCISE 6:

Use Javascript  
to calculate  $10+5$

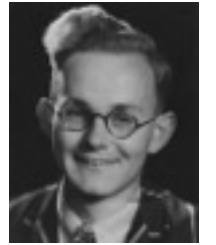


Pauze!

- [haxball.com](http://haxball.com)
- [agar.io](http://agar.io)
- [slither.io](http://slither.io)
- [curvefever.io](http://curvefever.io)
- Telegramautootje
- kattenfilmpjes
- ?



7



31



**JQUERY**



# Variables

```
var balloon = $(".balloon");  
bubble = balloon.find(".bubble");
```

**:{) Codaisseur**

# Keywords

```
var balloon = $(".balloon");  
bubble = balloon.find(".bubble");
```

**:{) Codaisseur**

10 min

## EXERCISE 7:

Select the balloon and  
call `.clone()` on it

# REPEAT

Until you are done

# Loop

```
var balloon = $(".balloon");  
var body = $("body");  
for(var i=0; i<10; i++){  
    var copy = balloon.clone();  
    copy.appendTo(body);  
}
```

**:{) Codaisseur**

# LIKE A BOX

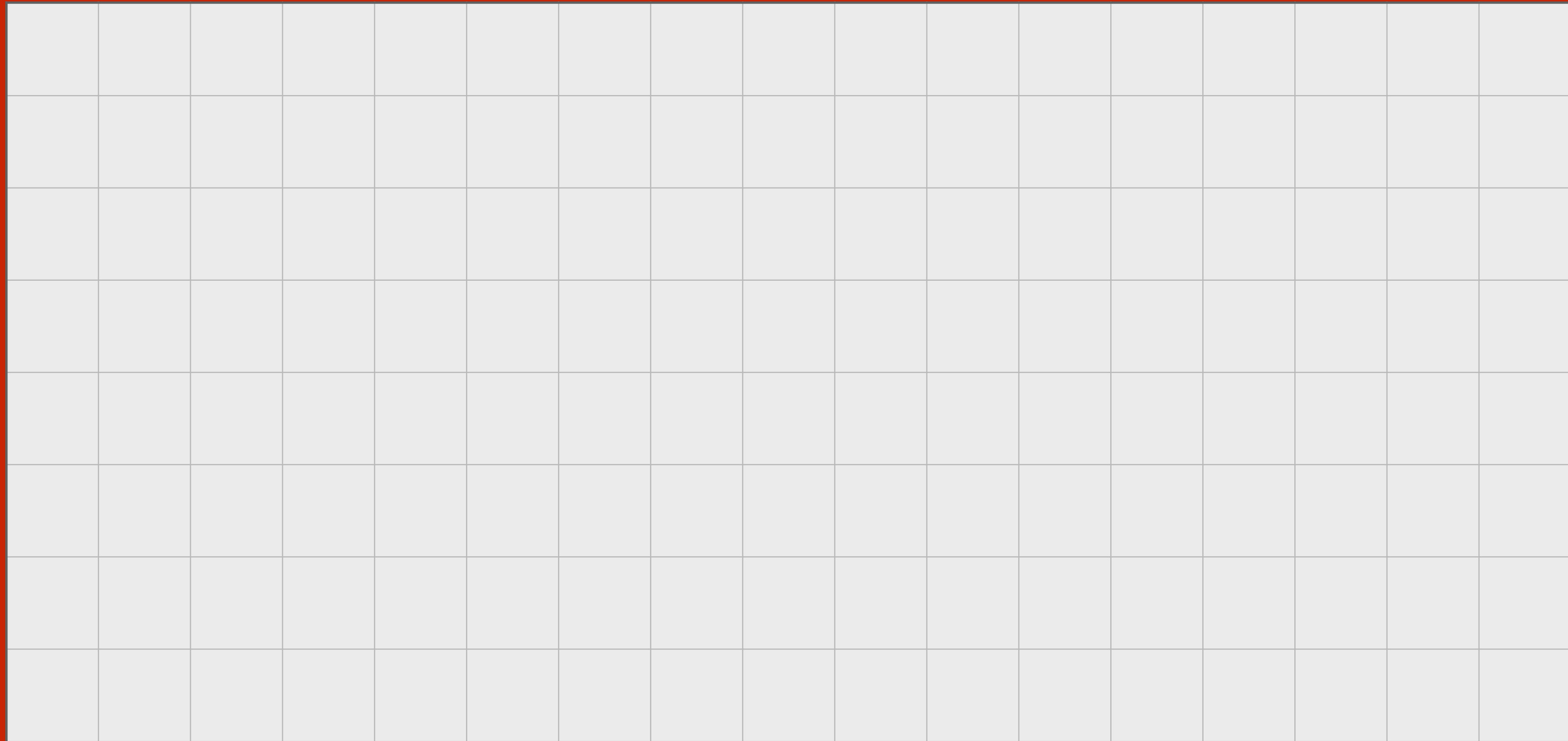
Put it away for later use

15 min

## EXERCISE 8:

**Create 10 balloons  
with loops**

# Position on the screen with CSS





# Positioning

```
.balloon {  
    position: absolute;  
    bottom: 0;  
}
```

**:{) Codaisseur**

15 min

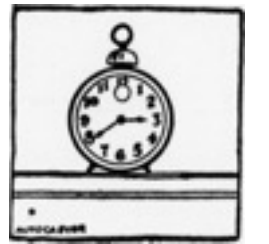
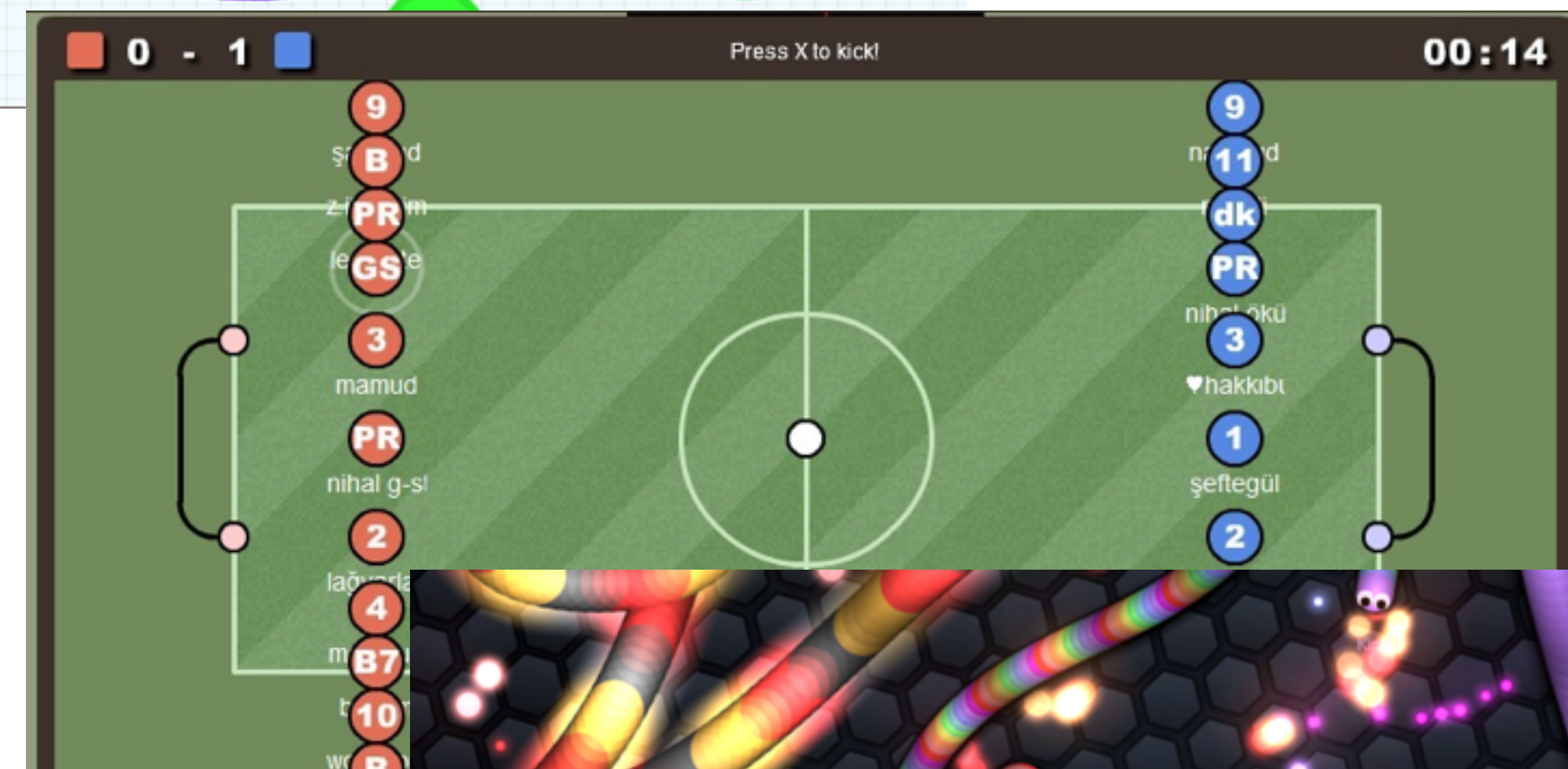
## EXERCISE 9:

**Position 10 balloons  
in a straight line**

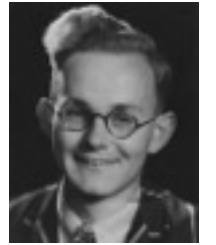


# Pauze!

- [haxball.com](http://haxball.com)
- [agar.io](http://agar.io)
- [slither.io](http://slither.io)
- [curvefever.io](http://curvefever.io)
- Telegramautootje
- kattenfilmpjes
- ?



7



31



# Callback

Method as argument of method

```
balloon.click(function(){  
    $( this ).remove();  
});
```

# this

evaluates to the value of the ThisBinding  
of the current **execution context**

20 min

## EXERCISE 10:

**Pop some balloons  
and keep score**

# Animate

```
var balloon = $( ".balloon" );  
balloon.animate({bottom: "100%"}, 8000);
```

**:{) Codaisseur**



15 min

# EXERCISE 11:

# Make em float

# Events

```
var balloon = $( ".balloon" );
```

```
balloon.click(function() {  
    $( this ).remove();  
});
```

**:{) Codaisseur**

# EXERCISE 12:

Put it online

**klik rechtsonder op [export] en  
upload dit op [bitballoon.com](https://bitballoon.com)**

**DONE!**

# DONE?

- **Show counter in the center of the screen after all balloons are offscreen**
- **Change the color of the balloons**
- **Add Sound**
- **Advanced animation**