

ICT-Start – Session 01

Module 4: Client-Side Scripting II Modular Programming

A) Modular Programming

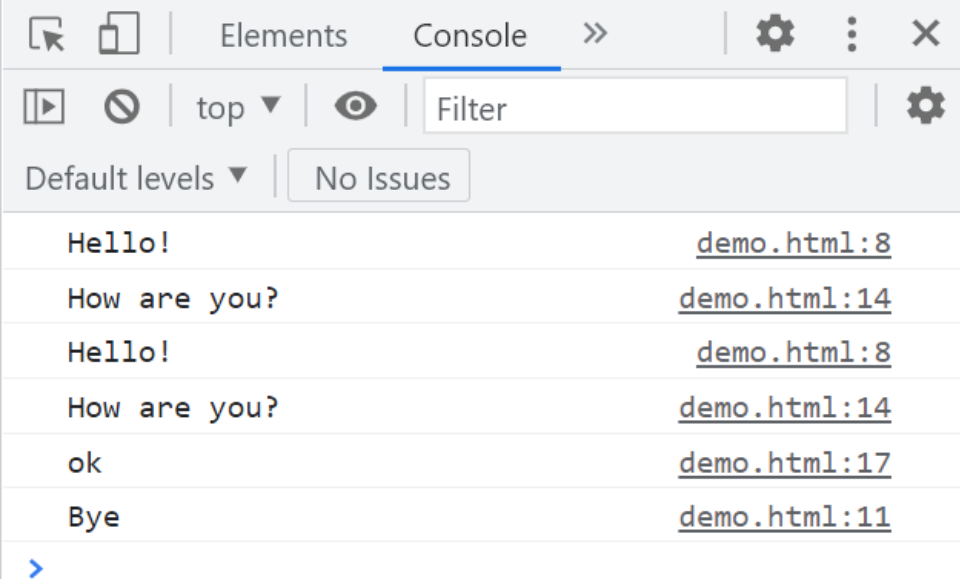
December 2022

Modular Programming - Intro

Modular Programming - Intro

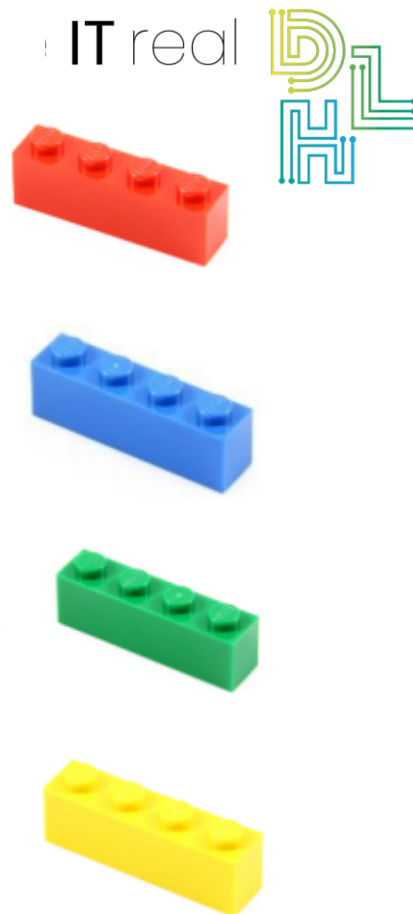
- Until now, we only used functions to evaluate events
- However, we can call functions manually without the need of an event!
- We did this actually several times...
 - `prompt()`
 - `console.log()`
 - `alert()`

Example



```
<title></title>  
<script type="text/javascript">  
  function hello(){  
    console.log("Hello!");  
  }  
  function bye(){  
    console.log("Bye");  
  }  
  function ask(){  
    console.log("How are you?");  
  }  
  function ok(){  
    console.log("ok");  
  }  
</script>
```

```
hello();  
ask();  
hello();  
ask();  
ok();  
bye();
```



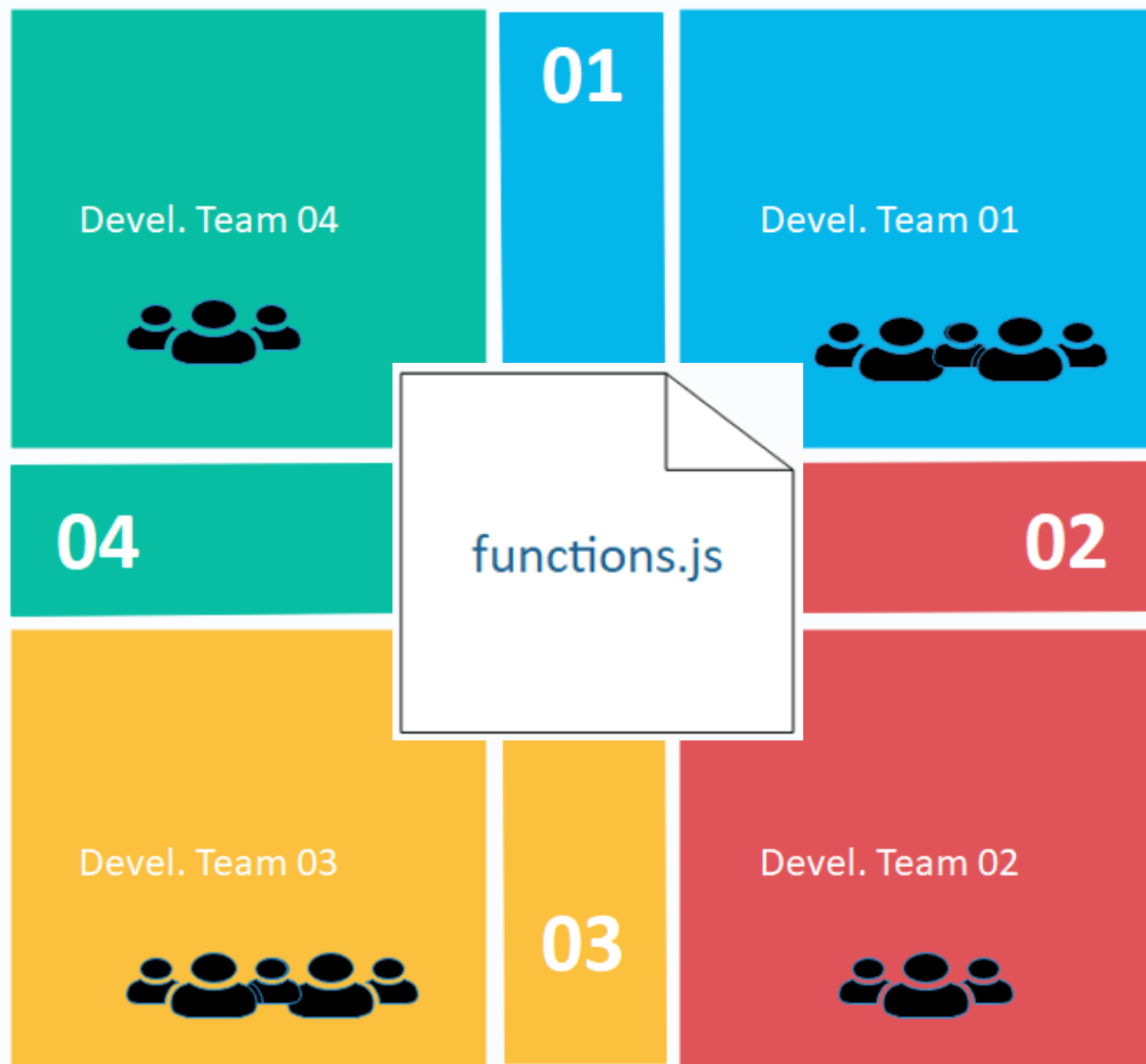
A better approach



- We have used functions to subdivide our program into smaller **subroutines**, making the new code more orderly, clear and structured.
- These are like **Lego bricks**, since they can be easily combined in many different ways.
- In addition, the code is also much easier to **read** because you can see from the function names what is actually executed.
- This is a very important aspect when you are working within a **development team** and sharing JavaScript files among each other. Because the code is more readable it is more understandable and accessible to every member of the team.

Working in a team

- Since we know how to encapsulated code in functions, how can we share the code between members in a team?
- **Hint:** How we did it with CSS?



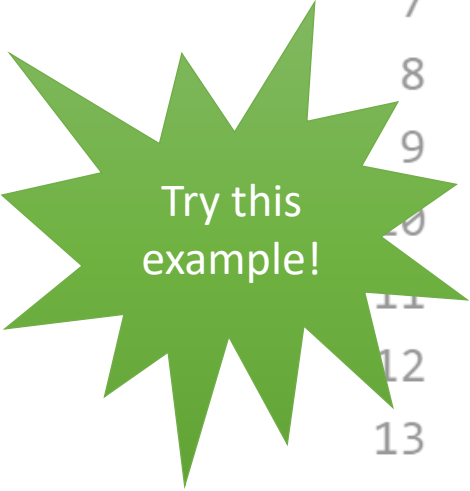
demo_outsourcing

> js

outsourced.html

```
1 <html>
2   <head>
3     <meta charset="utf-8">
4     <title></title>
5     <script type="text/javascript"
6       • src="js/functions.js"></script>
7     <script type="text/javascript">
8       hello();
9       ask();
10      hello();
11      ask();
12      ok();
13      bye();
14    </script>
15  </head>
16  <body>
17  </body>
18 </html>
```

```
1 function hello(){
2   console.log("Hello!");
3 }
4 function bye(){
5   console.log("Bye");
6 }
7 function ask(){
8   console.log("How are you?");
9 }
10 function ok(){
11   console.log("ok");
12 }
13
```



Try this
example!

Important note: In the script tag where you load the JS file, you cannot write JS-Code in between the tags. You can either load a JS file or write directly between the tags, but not both at the same time! That's why we need 2 script tags!

Exercises