

# L01 Types of String, Hoisted, Currying, Scopes, IIFE, Lambda Calculus, ...

`document.writeln("something")`: to write onto page

JavaScript functions:

- JavaScript doesn't need a Compiler
- 2 phase interpreter: first phase only locates space for variables and object, second phase initialises
- can't overload method, they get overwritten, "last one wins"
- no explicit return in functions returns undefined
- default parameter can be initialised in the method parameter list with e.g. `arg = 0`
- functions can be stored into const
- currying = transforms function with multiple arguments into a sequence of nesting functions
- hoisted: can be used before declaration like function, const is not hoisted
- three types of string: "it's time" and 'he said "hi"' and `x ist \${x}`  
`String(/hdghsj/)`  
(`typeof String = function`)
- two scopes:
  - global: alles zugreifbar (im browser: window)
  - function: variables are local to the enclosed function

```
let x = ...   mutable, local scope  
const x = ... immutable*, local scope
```

- IIFE: immediately invoked function expression

```
((() => {  
    let x = 1;  
    document.writeln(String(x));  
})());
```

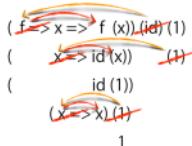
- Lambda Calculus

alpha: renames variables,  $x \Rightarrow x$  is alpha equivalent to  $y \Rightarrow y$

beta: runs function,  $(x \Rightarrow x+1)(5)$ , beta reduces to 6

eta: reduction,  $x \Rightarrow \text{foo}(x)$ , eta reduces the unnecessary stuff to foo

Beta Reduktion II



Eta Reduktion II

