

Web Programming

Week 2

Prof. D. König

Be Empowered

"Aufklärung ist der Auszug des Menschen aus seiner selbstverschuldeten Unmündigkeit. Unmündigkeit ist der Mangel, sich seines eigenen Verstandes ohne die Hilfe eines anderen zu bedienen. Selbstverschuldet ist diese Unmündigkeit, wenn sie nicht auf einem Mangel an Verstand sondern auf einem Mangel an Mut und Entschlusskraft beruht."

Immanuel Kant

Retrospective

JS Goodie

What we did

Catching up with the snake game

Questions

Agenda

Improving the game

Core topic: JavaScript Scopes

Lambda Calculus: Brain training

Quiz

Practice

Completing Snake

Improving the tests

Simulate motion & gravity (ball)



JavaScript Scopes

global window (in Browser)

function no matter where
defined, variables are
local to the enclosing
function (lambda)

JavaScript Variables

in scope after first use

~~x = ... mutable, global scope~~

~~var x = ... mutable, "hoisted" scope~~

let x = ... mutable, local scope

const x = ... immutable*, local scope

IIFE

immediately invoked function expression

```
function foo() {···}; foo()
```

```
(function foo() {···}) ()
```

```
(function() {···}) ()
```

```
( () => {···}) ()
```


Lambda Calculus

α alpha: Rename parameter

β beta: Apply argument


η eta: Cancel parameter

Alpha Translation


$\text{const id} = x \Rightarrow x$

$\text{const id} = y \Rightarrow y$

Beta Reduktion




$$(\lambda x. f x) (\lambda x. x) (\lambda x. x) (1)$$



$$(\lambda x. x) (\lambda x. x) (\lambda x. x) (1)$$

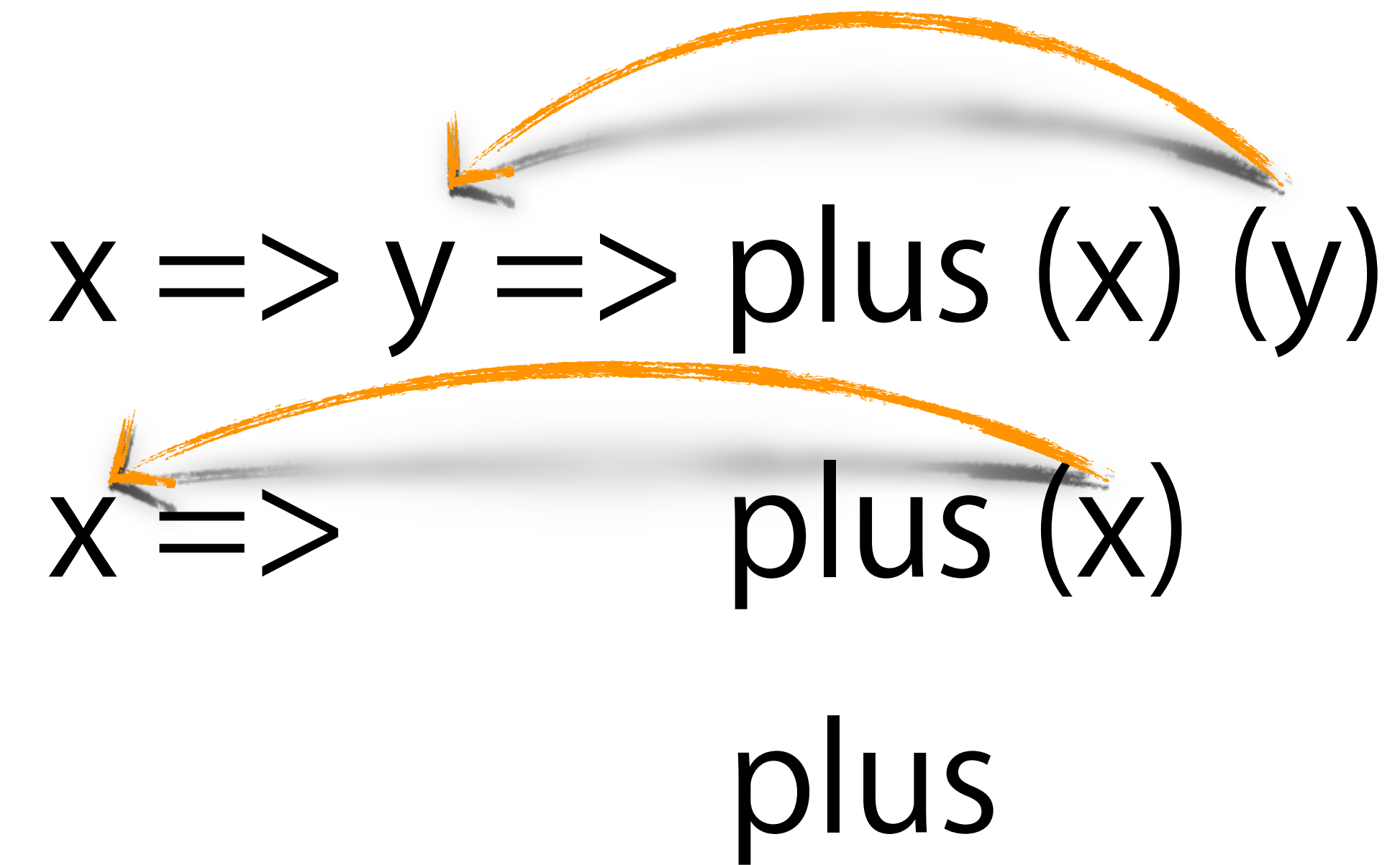
$$(\lambda x. x) (\lambda x. x) (\lambda x. x) (1)$$



$$(\lambda x. x) (\lambda x. x) (\lambda x. x) (1)$$

1

Eta Reduktion



Homework

watch Gabriel Lebec (~1:40)

Fundamentals of Lambda Calculus &
Functional Programming in JavaScript,
Parts I and II.

[https://www.youtube.com/watch?
v=3VQ382QG-y4](https://www.youtube.com/watch?v=3VQ382QG-y4)