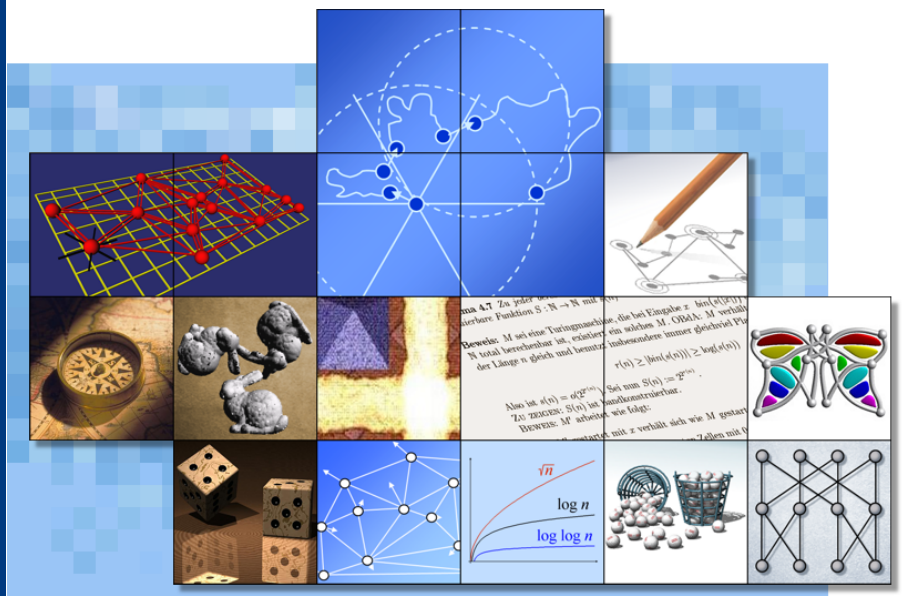
HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

EScript

Short Presentation of a Scripting Language



May 11, 2012

Benjamin Eikel

Overview

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

1 Introduction

2 Data Types

3 Control Structures

4 Examples

What is EScript?

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

EScript . . .

- is a interpreted, object-oriented scripting language.
- has a similar syntax to C.
- was developed to use C++ objects from scripts easily.
- is released under a free software license.
- is available from `http://escript.berlios.de/`.

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

- EScript files should have the extension `.escript`.
- A simple script:

```
out("Hello, world!\n");
```

Simple Types

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Number Examples: `1`, `27.4`, `0x1a`, `25 / 5`, `3 + 4`

String Examples: `"an"`, `'example'`, `"hello"+ ' ' + "world"`

Bool **`true`** or **`false`**

Void **`void`**

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

var

```
/*
    Declaring a variable is done using the keyword
    "var", an identifier, an equation sign, and an
    expression on the right side. The type of the
    variable is deduced from the expression on the
    right side.
 */
var xPos = 500 - 80 / 2;

// The variable "message" will be of type String
var message = "Please click the button";

// Dynamically change the type to Number
message = 5;
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

fn

```
var square = fn(num) {
    return num * num;
};
var a = square(5);
var b = square(4.2);
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Array

```
var numbers = [3, 23, 7, 3, 100, 1, 35];
var colors = ["red", "green", "blue"];
```

Map

```
var fruits = {
    "lemon" : "yellow",
    "cherry" : "red"
};
fruits["apple"] = "green";
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

ExtObject

```
var car = new ExtObject();
car.color := "red";
car.speed := 190;
car.outputDesc := fn() {
    out("This is a ", this.color, " car ");
    out("with top speed ", this.speed, ".\n");
};

...

car.speed = 185;
car.outputDesc();
```

Output: This is a red car with top speed 185.

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Type

```
var Shape = new Type();
Shape.color := "white";

// New type that is derived from Shape
var Polygon = new Type(Shape);
Polygon.numVertices := 3;

// New type that is derived from Shape
var Circle = new Type(Shape);
Circle.radius := 0;

var circle = new Circle();
circle.color = "red";
circle.radius = 5;
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

if

```
var result = /* some function */;
if(result) {
    out("Success");
} else {
    out("Failure");
}

var num = /* some number */;
if(num < 0) {
    out("Too small");
} else if(num >= 0 && num <= 100) {
    out("Range okay");
} else {
    out("Too large");
}
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

? (conditional operator)

```
var num = /* some number */;
var positive = (num > 0) ? true : false;
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

while

```
var tasks = [/* some tasks */];
while(!tasks.empty()) {
    var firstTask = tasks.front();
    tasks.popFront();
    // do something with first task
}
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

for

```
var sum = 0;
for(var i = 0; i < 100; ++i) {
    sum += i;
}
out("Sum of numbers: ", sum, "\n");
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

foreach

```
var chars = ["a", "c", "k", "b", "d", "x", "j"];
foreach(chars as var c) {
    if(c == "x") {
        out("Character \"x\" found.");
        break;
    }
}
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Factorial: $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$ $0! = 1$

Implementation

```
var factorialRecursive = fn(Number n) {
    if(n == 0) {
        return 1;
    }
    return thisFn(n - 1) * n;
};
var factorialIterative = fn(Number n) {
    var product = 1;
    for(var i = 2; i <= n; ++i) {
        product *= i;
    }
    return product;
};
```

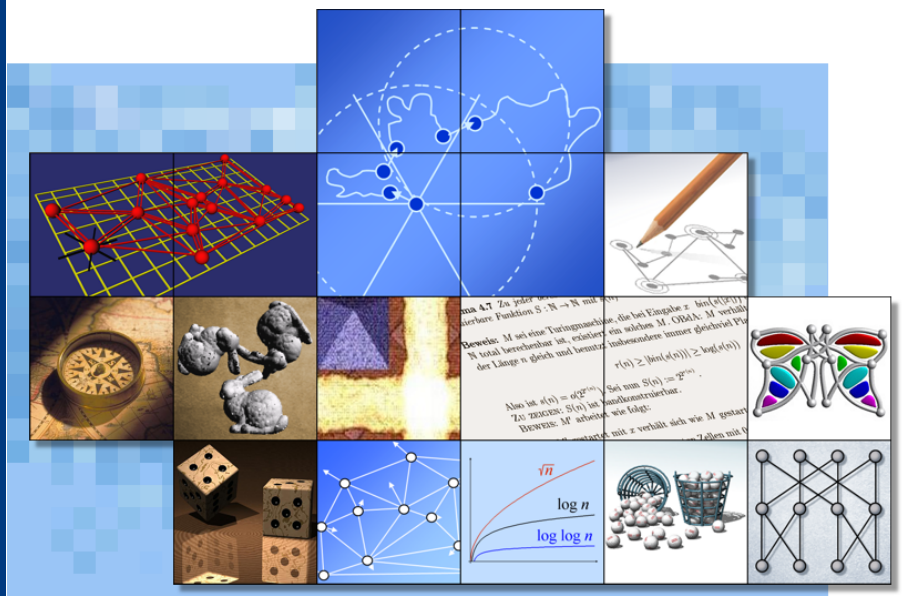
HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Implementation

```
var Player = new Type();
Player.x := 0;
Player.y := 0;
var movePlayer = fn(player, Number dx, Number dy) {
    player.x += dx;
    player.y += dy;
};
var printPos = fn(player) {
    out("Player position: (", player.x);
    out(", ", player.y, ")\n");
};

var playerA = new Player();
movePlayer(playerA, 5, 7);
printPos(playerA);
```

Thank you for your attention!



Benjamin Eikel

Heinz Nixdorf Institute
& Department of Computer Science
University of Paderborn

Address: Fürstenallee 11
33102 Paderborn
Germany

Phone: +49 5251 60-6452
Fax: +49 5251 60-6482
E-mail: eikel@upb.de
Web: http://wwwhni.upb.de/en/alg/

Benjamin Eikel holds a scholarship of the
International Graduate School
Dynamic Intelligent Systems