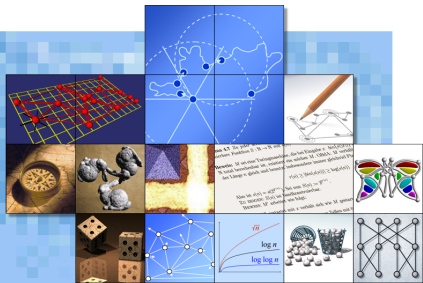


EScript

Kurzvorstellung einer Skriptsprache

13. Februar 2012

Benjamin Eikel





- 1 Einführung
- 2 Datentypen
- 3 Kontrollstrukturen
- 4 Beispiele

EScript ...

- ist eine interpretierte, objektorientierte Skriptsprache.
- hat eine ähnliche Syntax wie C.
- wurde entwickelt, um C++-Objekte einfach in Skripten verwenden zu können.
- ist unter einer freien Softwarelizenz veröffentlicht.
- ist erhältlich unter `http://escript.berlios.de/`.

- EScript-Dateien sollten die Endung `.escript` haben.
- Ein einfaches Skript:

```
out("Hallo Welt!\n");
```

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

String Beispiele: `"ein"`, `'beispiel'`, `"hallo" + "welt"`

Bool `true` oder `false`

Void `void`

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;
```

fn

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

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";
```


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.

Type

```
var Shape = new Type();  
Shape.color := "white";  
  
// Neuer Typ, der von Shape erbt  
var Polygon = new Type(Shape);  
Polygon.numVertices := 3;  
  
// Neuer Typ, der von Shape erbt  
var Circle = new Type(Shape);  
Circle.radius := 0;  
  
var circle = new Circle();  
circle.color = "red";  
circle.radius = 5;
```

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");
}
```

while

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

for

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

foreach

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

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n \quad 0! = 1$$

Implementierung

```
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;  
};
```

Implementierung

```
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);
```




Vielen Dank für Ihre Aufmerksamkeit!

Benjamin Eikel
Heinz Nixdorf Institut
& Institut für Informatik
Universität Paderborn

Adresse: Fürstenallee 11
33102 Paderborn
Deutschland

Telefon: +49 5251 60-6452
Fax: +49 5251 60-6482
E-Mail: eikel@upb.de
Web: <http://www.hni.upb.de/alg/>

Benjamin Eikel ist Stipendiat der

 **International Graduate School**
Dynamic Intelligent Systems

