basil.js / Cheatsheet

v0.2

basil.js

An attempt to port the spirit of the Processing visualization language to Adobe Indesign. Cheatsheet by Ludwig Zeller.

www.basiljs.ch

Color

```
b.color(255,0,0,"red"); // adds a color to the swatches named "red"
b.fill(255); // or b.fill(255,255,255); or b.fill(mySwatch);
b.fillTint(255); // like b.fill(); they all accept either C,M,Y,K or R,G,B or gray
b.noFill(); // turns off filling in all following commands
b.noStroke(); // turns off strokes in all following commands
b.stroke(255); // like b.fill() it is used to set a color
```

Drawing

```
b.ellipse(x,y,w,h); // draws an ellipse or circle at given position and size
b.line(x1,y1,x2,y2); // draws a line from one point to another
b.rect(x,y,w,h); // like b.ellipse() but for rectangles
b.ellipseMode(); // centering b.CORNER, b.CORNERS, b.CENTER
b.rectMode(); // centering b.CORNER, b.CORNERS, b.CENTER
b.strokeWeight(1); // set the strokes in the current unit
```

Transformation

```
b.itemX(obj, x); // positions the given object on the x axis
b.itemY(obj, y); // positions the given object on the y axis
b.itemWidth(obj, width); // sets the width for the given object
b.itemHeight(obj, height); // sets the height for the given object
b.itemPosition(obj, x, y); // repositions the given object
b.itemSize(obj, width, height); // resizes the given object
b.bounds(obj); // returns an object with width, height, left, right, top, bottom and
for text: baseline, xHeight as fields to describe the dimensions of the given object
b.height // the height of the page, use b.height / 2 to center vertically
b.width // the width of the page, use b.width/ 2 to center horizontically
```

Typography

```
b.stories(obj, func); // call a function on each story of object
b.paragraphs(obj, func); // call a function on each paragraph of object
b.lines(obj, func); // call a function on each line of object
b.words(obj, func); // call a function on each word of object
b.characters(obj, func); // call a function on each character of object
b.objectStyle("myStyle"); // return or create style with given name
b.characterStyle("myStyle"); // return or create style with given name
b.paragraphStyle("myStyle"); // return or create style with given name
b.text(txt,x,y,w,h); // create a textfield with given string and dimensions
b.textAlign(Justification.LEFT_ALIGN,
VerticalJustification.BOTTOM_ALIGN); // check Jongware
b.textFont("Helvetica","Bold"); // set default font
b.textKerning(70); // set default kerning
b.textLeading(96); // set default leading
b.textSize(72); // set default font size
b.textTracking(20); // set default tracking
b.typo(obj, "pointSize", 48); // change special parameters, check
Jongware for more information about available fields. obj can be a Document, Spread, Page,
Layer, Story, TextFrame or any Text object like Line, Paragraph, Word or Character.
```

Meta

```
b.doc(doc); // sets current document, without parameter it returns it
b.page(5); // returns the current page or sets it, also creates new pages
b.layer("myLayer"); // returns the current layer or sets it.
b.units(b.MM); // set the unit system to b.MM, b.IN, b.CM, b.PX, b.PT
b.guideX(200); // creates a new guide at the given position, also b.guideY();
b.selections(); // returns an array with all selected objects of the document
b.labels("myLabel"); // returns PageItems with that script label
b.go(); // starts basil.js... use b.loop() for the interactive environment
b.close(); // closes the currently focused document
```

Image

```
b.image(filename,x,y,w,h); // loads and adds picture from data folder
b.transformImage(img,x,y,w,h); // resizing pictures...
b.imageMode(); // choose between b.CORNER, b.CORNERS and b.CENTER
```

Basics

```
var myNumber = 5; // Integers...
var myNumber = 3.0; // Floats...
var myString = "Hello Basil"; / Strings...
var myLove = true; // and Booleans are all stored to the "var" type

// This is an inline comment
/* This is a block comment,
that can go over several lines */

function myFunction (param1, param2){
   // code within the brackets has access to param1 and param2 and will be executed...
}
myFunction(234, "Some String"); // You don't have to use b. for this
```

Control Structures

```
// for loops
for( var i = 0; i < 100; i++  )
{
  println( i ); // prints 100 numbers counting upwards
}
// if else conditionals
if ( a > 3 )
{
 // execute if a is bigger than 3
} else {
  // execute if a is smaller or equal to 3
}

a == b // equal
a === b // equal value and of the same data type... to be VERY sure
a != b // not equal
a > b // bigger than
a < b // smaller than
a >= b // bigger or equal
a <= b // smaller or equal

// You can combine statements with logical AND "&&" resp. the logical OR "||"
if (  a == b && b == c ) { ... } // AND
if (  a == b || b == c ) { ... } // OR
```

Math

```
b.abs(-5); // returns the absolute value, in this case it is 5
b.constrain(myNum,0,20); // clips myNum to 0 or 20 if it's lower or higher
b.map(val,0,1,2,5); // maps val between 2-5 if it moves between 0-1
b.lerp(min,max,val); // moves between min-max with the val between 0-1
+, -, *, / // addition, substraction, multiplication and division at your fingertips...
foo = foo + 5; // increases the variable foo by 5
foo += 5; // a short version of foo = foo + 5;
foo++; // even shorter version for foo = foo + 1;
b.noise(phase); // use phase to move through an harmonic noise space
b.random(50); // total chaos between 0-50. b.random(20,50); gives sth from 0-50
```

Output

```
b.println("Hello Basil"); // prints the message to the console
b.savePDF("export.pdf", true); // outputs a pdf next to the InDesign document file. The second parameter sets if the export dialogue should pop up. If false, the last used settings will be used again.
```