Code Convention

Regelwerk für das Erstellen von Quellcode

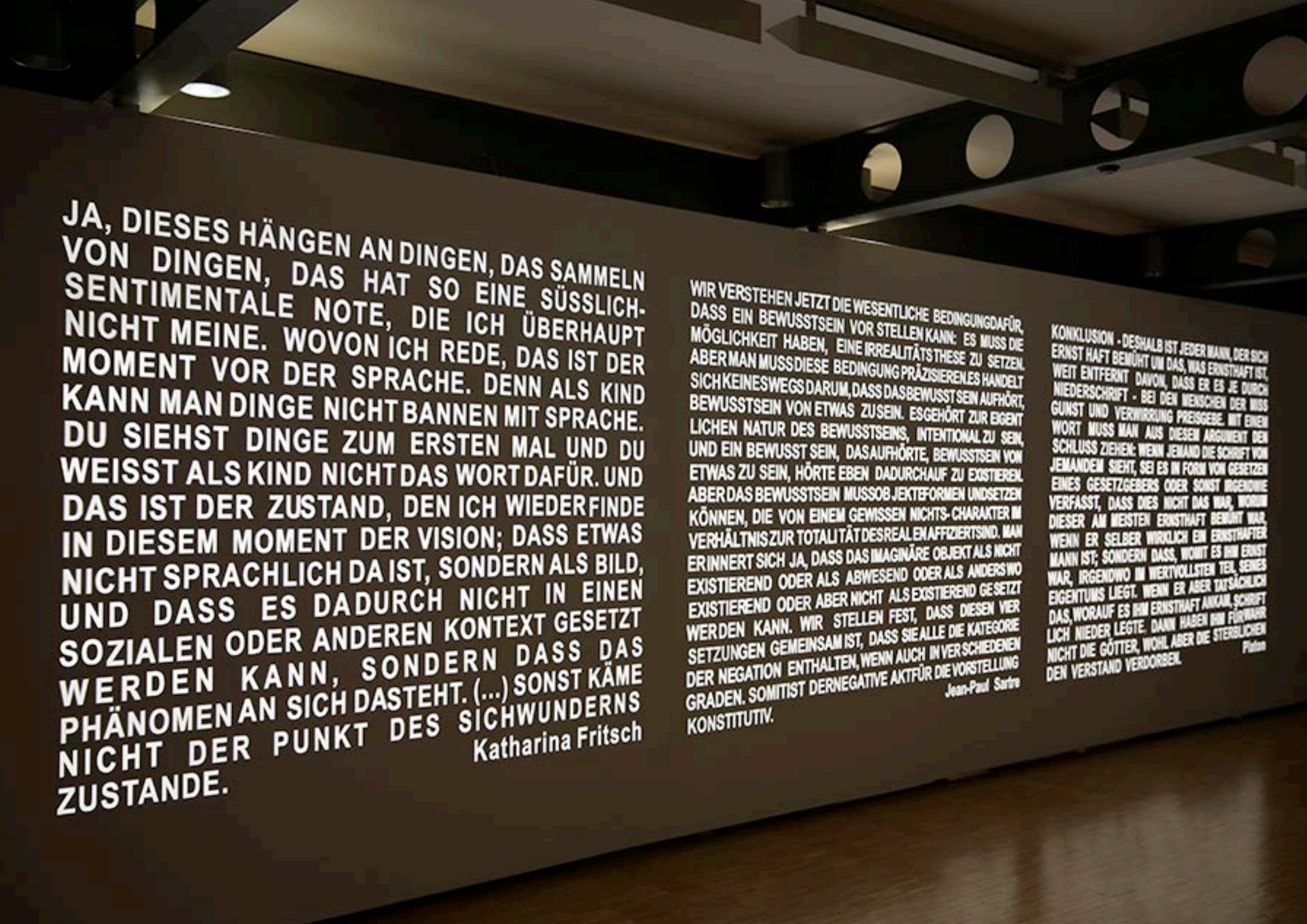
Christian Marienfeld | 2016

Code Convention

Regeln ?

Warum?

- lesbar
- verständlich
- wartbar


JA, DIESES HÄNGEN AN DINGEN, DAS SAMMELN VON DINGEN, DAS HAT SO EINE SÜSSLICH-SENTIMENTALE NOTE, DIE ICH ÜBERHAUPT NICHT MEINE. WOVON ICH REDE, DAS IST DER MOMENT VOR DER SPRACHE. DENN ALS KIND KANN MAN DINGE NICHT BANNEN MIT SPRACHE. DU SIEHST DINGE ZUM ERSTEN MAL UND DU WEISST ALS KIND NICHT DAS WORT DAFÜR. UND DAS IST DER ZUSTAND, DEN ICH WIEDER FINDE IN DIESEM MOMENT DER VISION; DASS ETWAS NICHT SPRACHLICH DA IST, SONDERN ALS BILD, UND DASS ES DADURCH NICHT IN EINEN SOZIALEN ODER ANDEREN KONTEXT GESETZT WERDEN KANN, SONDERN DASS DAS PHÄNOMEN AN SICH DASTEHT. (...) SONST KÄME NICHT DER PUNKT DES SICHWUNDERNS ZUSTANDE.
Katharina Fritsch
WIR VERSTEHEN JETZT DIE WESENTLICHE BEDINGUNG DAFÜR, DASS EIN BEWUSSTSEIN VOR STELLEN KANN: ES MUSS DIE MÖGLICHKEIT HABEN, EINE IRREALITÄTSTHESE ZU SETZEN. ABER MAN MUSS DIESE BEDINGUNG PRÄZISIEREN. ES HANDELT SICH KEINESWEGS DARUM, DASS DAS BEWUSSTSEIN AUFHÖRT, BEWUSSTSEIN VON ETWAS ZU SEIN. ES GEHÖRT ZUR EIGENTLICHEN NATUR DES BEWUSSTSEINS, INTENTIONAL ZU SEIN, UND EIN BEWUSST SEIN, DAS AUFHÖRTE, BEWUSSTSEIN VON ETWAS ZU SEIN, HÖRTE EBEN DADURCH AUF ZU EXISTIEREN. ABER DAS BEWUSSTSEIN MUSS OB JEKTEFORMEN UNDSETZEN KÖNNEN, DIE VON EINEM GEWISSEN NICHTS-CHARAKTER IM VERHÄLTNIS ZUR TOTALITÄT DES REALEN AFFIZIERT SIND. MAN ERINNERT SICH JA, DASS DAS IMAGINÄRE OBJEKT ALS NICHT EXISTIEREND ODER ALS ABWESEND ODER ALS ANDERSWO EXISTIEREND ODER ABER NICHT ALS EXISTIEREND GESETZT WERDEN KANN. WIR STELLEN FEST, DASS DIESEN VIER SETZUNGEN GEMEINSAM IST, DASS SIE ALLE DIE KATEGORIE DER NEGATION ENTHALTEN, WENN AUCH IN VERSCHIEDENEN GRADEN. SOMIT IST DER NEGATIVE AKT FÜR DIE VORSTELLUNG KONSTITUTIV.
Jean-Paul Sartre
KONKLUSION - DESHALB IST JEDER MANN, DER SICH ERNST HAFT BEMÜHT UM DAS, WAS ERNSTHAFT IST, WEIT ENTFERNT DAVON, DASS ER ES JE DURCH NIEDERSCHRIFT - BEI DEN MENSCHEN DER MISS GUNST UND VERWIRRUNG PREISGEBE. MIT EINEM WORT MUSS MAN AUS DIESEM ARGUMENT DEN SCHLUSS ZIEHEN: WENN JEMAND DIE SCHRIFT VON JEMANDEN SIEHT, SEI ES IN FORM VON GESETZEN EINES GESETZGEBERS ODER SONST IRGENDWIE VERFASST, DASS DIES NICHT DAS WAR, WORUM DIESER AM MEISTEN ERNSTHAFT BEMÜHT WAR, WENN ER SELBER WIRKLICH EIN ERNSTHAFTER MANN IST; SONDERN DASS, WOMIT ES IHM ERNST WAR, IRGENDWO IM WERTVOLLSTEN TEIL SEINES EIGENTUMS LIEGT. WENN ER ABER TATSÄCHLICH DAS, WORAUF ES IHM ERNSTHAFT ANKAM, SCHRIFT LICH NIEDER LEGTE. DANN HABEN IHN FÜRWAHR NICHT DIE GÖTTER, WOHL ABER DIE STERBLICHEN DEN VERSTAND VERDORBEN.
Platon

QUELLCODE

- formal
- strukturiert

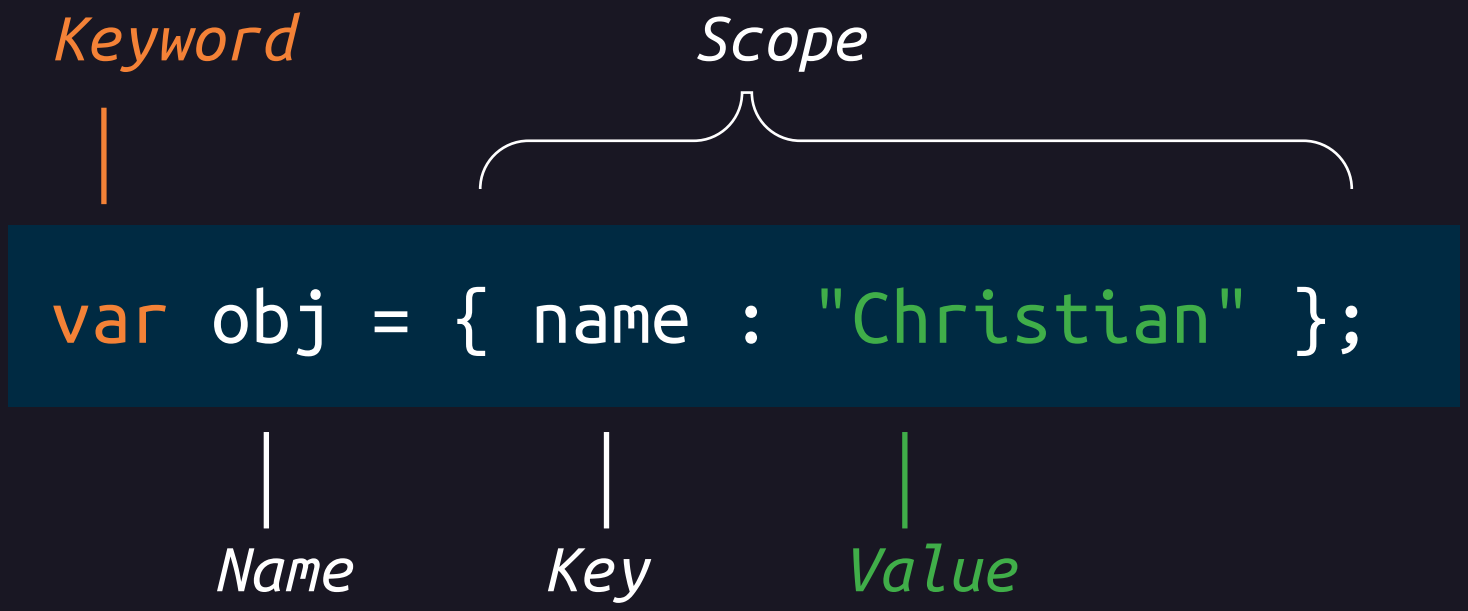
Unabhänging vom Nuten oder Inhalt!


AEG

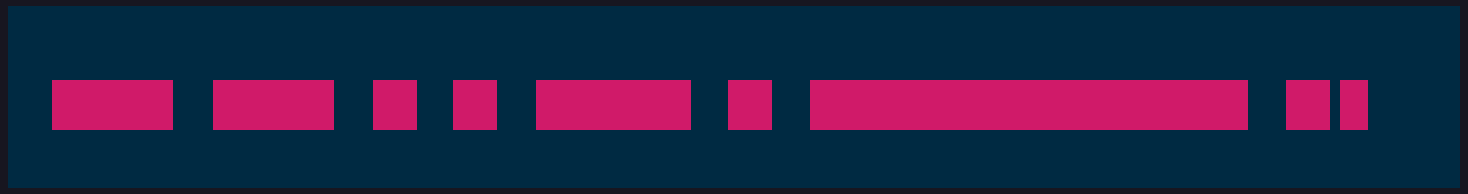

TULS
182197
SE 2106

Syntax JavaScript

Syntax JavaScript

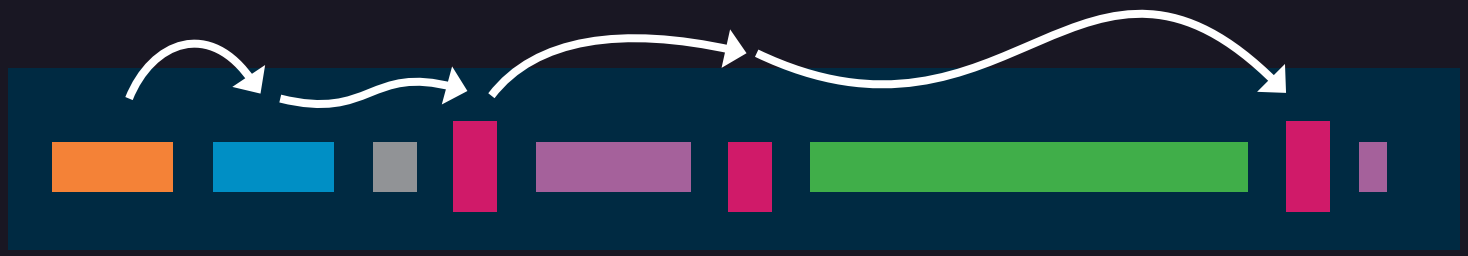


Leerraum



```
var obj = { name : "Christ
```

Lesbarkeit



```
var obj = { name : "Christian" };
```

Eins, Zwei, Drei... Vorbei!

1. ```
   var obj={name:"Christian"};
   ```

2. ```
   var obj = { name : "Christian" };
   ```

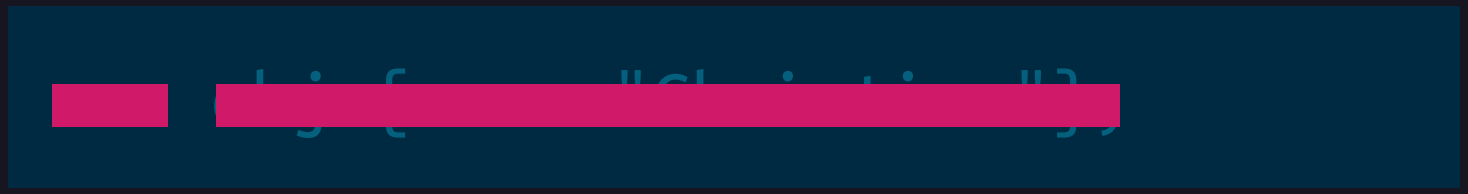
3. ```
   var obj = {name:"Christian"};
   ```
   ```

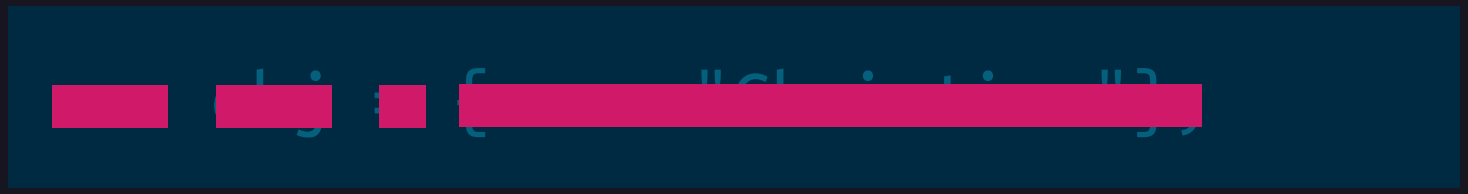
...Vorbei!

2.

```
var obj = { name : "Christian" };
```

Leerraum





Identifier Names

1.
```
var firstName = "Christian";
```

2.
```
var first_name = "Christian";
```

3.
```
var FirstName = "Christian";
```

Identifier Names

1.
```
var firstName = "Christian";
```

2.
```
var first_name = "Christian";
```

3.
```
var FirstName = "Christian";
```


longCamelCaseNames

Operators

1.
```
var x = y+z;
```

2.
```
var x=y + z;
```

3.
```
var x = y + z;
```

Operators

3.

```
var x = y + z;
```

Operators Love Space

```
=  +  -  *  /
```

Und Komma ?

Listen / Array

1.
```
var obst = [ "Apfel","Birne","Orange" ];
```

2.
```
var obst = [ "Apfel", "Birne", "Orange" ];
```

3.
```
var obst = [ "Apfel" , "Birne" , "Orange" ];
```

Listen / Array

2.

```
var obst = [ "Apfel", "Birne", "Orange" ];
```

Function / Scopes

1.
```
function removeOne (num){return num-1;}
```

2.
```
function removeOne( num ) { return num-1; }
```

3.
```
function removeOne(num){ return num-1; }
```

Function / Scopes

Function / Scopes

1.
```
function removeOne( num ) {
    return num-1;
}
```

2.
```
function removeOne( num )
{
return num-1;
}
```

3.
```
function removeOne
( num ) {
    return num-1;
}
```

Function / Scopes

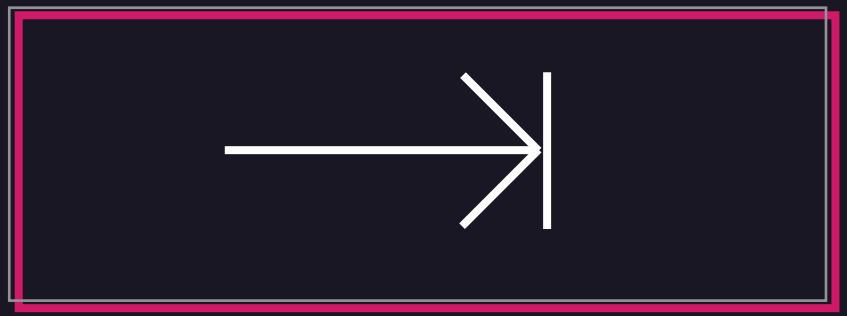
```
function removeOne(num) {
    return num-1;
}
```

```
function removeOne ( num ) {
    return num-1;
}
```

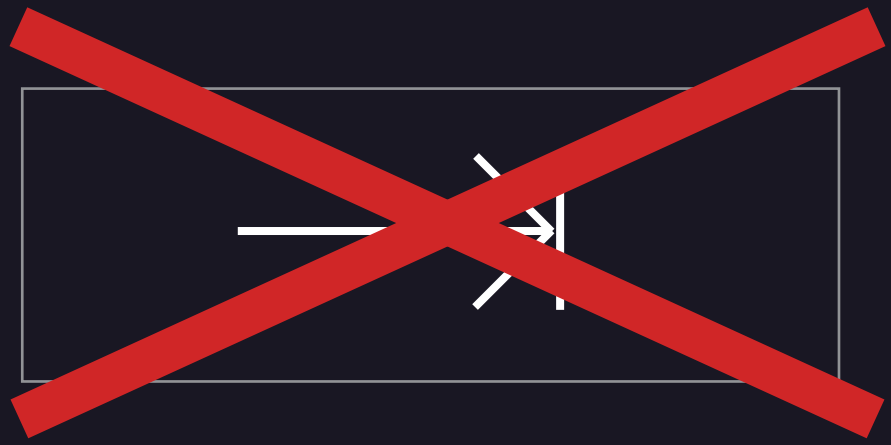
```
function removeOne ( num )
{
    return num-1;
}
```

Tab

```
function removeOne(num) {
    if (num <= 10) {
        return num - 1;
    } else {
        return num - 10;
    }
    return false;
}
```



No Tab!



use 4 spaces !!!

... search plugin or script

```
var myApp = {
                version: 1,
                users: ["Christian", "Eva"],
                getUsers: function () {
                    return a.users;
                },
                setUsers: function (arr) {
                    if (arr) {
                        a.users = arr;
                        a.version++;
                        return true;
                    }
                    return false;
                }
            };
```

Simple Statement

```
var values = ["Volvo", "Saab", "Fiat"];

var person = {
    firstName: "John",
    lastName: "Doe",
    age: 50,
    eyeColor: "blue"
};
```

- Always end a simple statement with a semicolon
- Use multiple lines if the statement is too complex

Complex Statement

```
function toCelsius(fahrenheit) {
    return (5 / 9) * (fahrenheit - 32);
}
```

- Put the opening bracket at the end of the first line.
- Use one space before the opening bracket.
- Put the closing bracket on a new line, without leading spaces.
- Do not end a complex statement with a semicolon.

Complex Statement

```
function toCelsius( fahrenheit ) {
    return (5 / 9) * (fahrenheit - 32);
}
```

```
for ( i = 0; i < 5; i++ ) {
    x += i;
}
```

```
if ( time < 12 ) {
    greeting = "Good Night";
} else {
    greeting = "Good Morning";
}
```

JavaScript Object Rules

```
var person = {
    firstName: "John",
    lastName: "Doe",
    age: 50,
    eyeColor: "blue"
};
```

- Place the opening bracket on the same line as the object name.
- Use colon plus one space between each property and its value.
- Use quotes around string values, not around numeric values.
- Do not add a comma after the last property-value pair.
- Place the closing bracket on a new line, without leading spaces.
- Always end an object definition with a semicolon.

JavaScript Object Rules

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

Line Length < 80

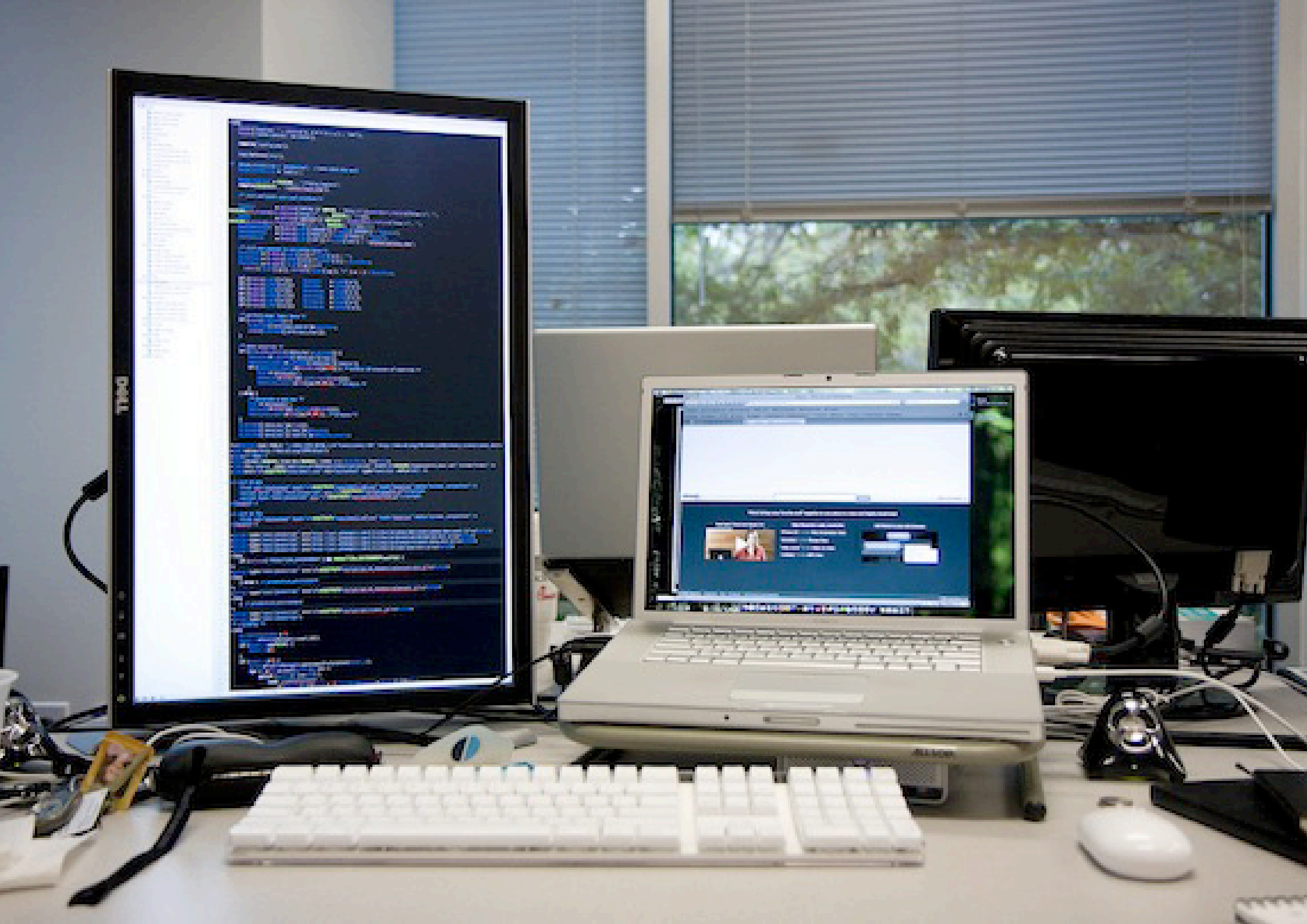
```
document.getElementById("demo").innerHTML =
    "Hello Dolly.";
```

- Break it after an operator or a comma.

Screen Size

```
var code = true;

function toLongForScreen() {
    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
    ...
} // Ende der function
```



JavaScript Kommentare

/* Kommentar */

```
/* der Kommentar beginnt hier ...
openPage(4);
...und endet mit einem */
```

// Kommentar

```
// nur diese Zeile ist ein Kommentar
openPage(4);
// die Zeile davor wird ausgefuehrt
```

Kommentare

/* Kommentar */

```
WAS macht der Code?

    -> Was nimmt er an?
    <- Was gibt er zurück?
```

// Kommentar

```
WAS und WARUM schreiben wir dies?
```

Kommentare

/* Kommentar */

```
/**
* Returns the n-th color of all colors
*
* @param int n number of the color, starts with 0.
* @return string color
*/
function getColor(n) {
    if ( colors[n] ) {
        return colors[n];
    }
    return false;
}
```

Kommentare

// Kommentar

```
// get third color because red/3 is the most common color
var defaultColor = getColor(3);
```

Naming Conventions

```
var firstName = "Christian";

var PI = 3.14;
```

- variable and function names -> **camelCase**
- global variables -> **UPPERCASE**

Naming Conventions

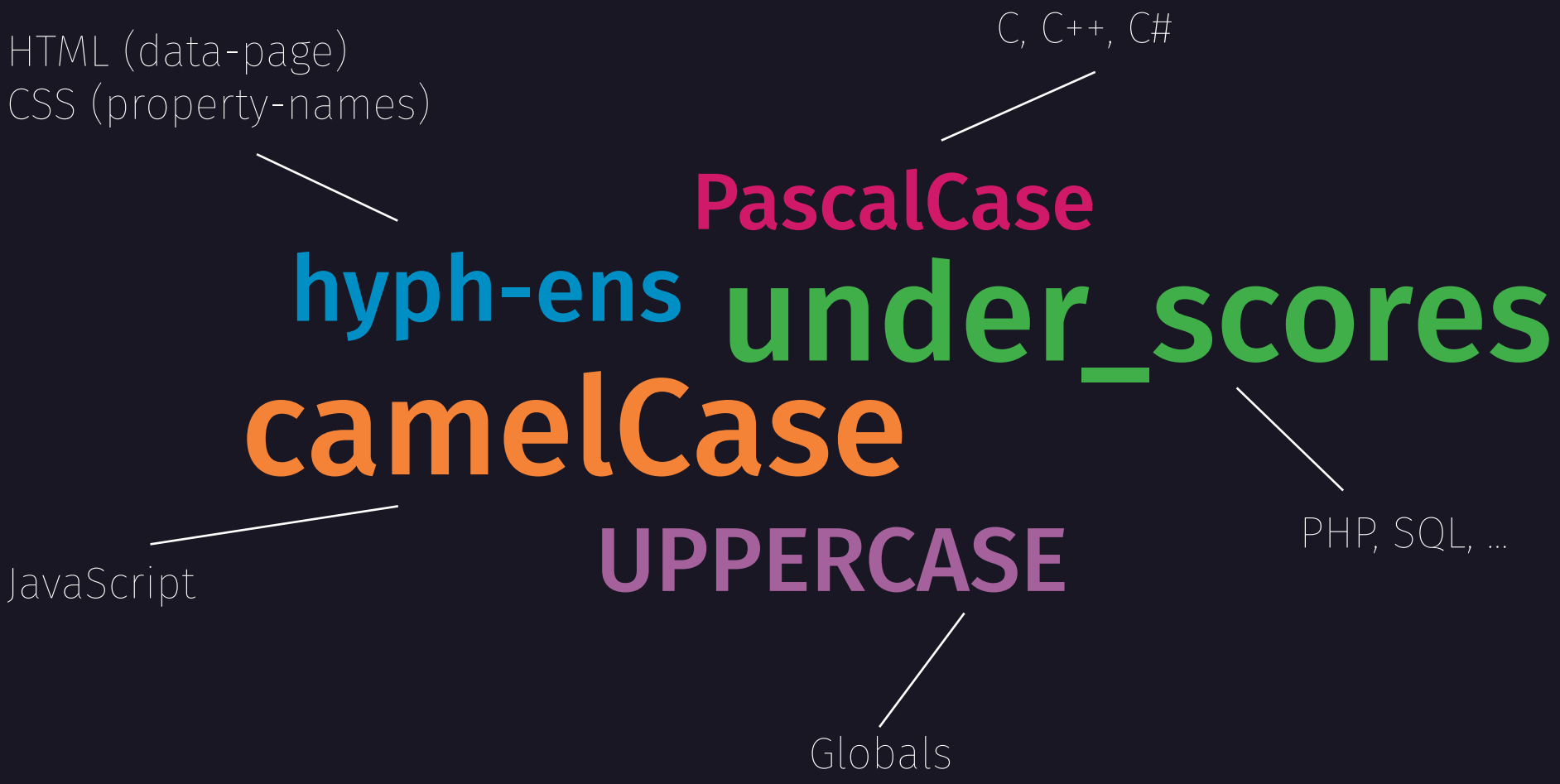
PascalCase

hyph-ens under_scores

camelCase

UPPERCASE

Naming Conventions



Naming Conventions

```
var obj = getElementById("Demo");
```

!=

```
var obj = getElementById("demo");
```

Naming Conventions

```
London.jpg
```

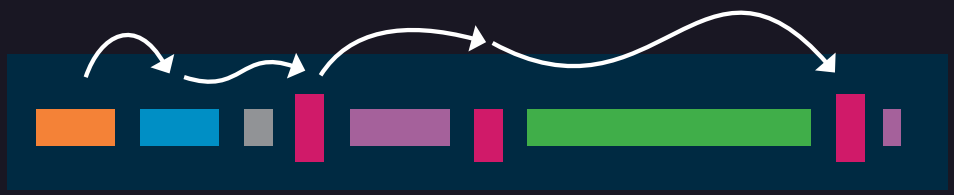
!=

```
london.jpg
```

!=

```
london.JPG
```

Zusammenfassung



```
function toCelsius( fahrenheit ) {
    return (5 / 9) * (fahrenheit - 32);
}
```

```
for ( i = 0; i < 5; i++ ) {
    x += i;
}
```

```
if ( time < 12 ) {
    greeting = "Good Night";
} else {
    greeting = "Good Morning";
}
```

```
var values = ["Volvo", "Saab", "Fiat"];

var person = {
    firstName: "John",
    lastName: "Doe",
    age: 50,
    eyeColor: "blue"
};
```

```
var myApp = {
    version: 1,
    users: ["Christian", "Eva"],
    getUsers: function () {
        return a.users;
    },
    setUsers: function (arr) {
        if (arr) {
            a.users = arr;
            ...
```

Quellen

http://www.w3schools.com/js/js_conventions.asp

http://www.php-tagebuch.de/dokumentation-wie-kommentiere-ich-richtig/

http://images.fotocommunity.de/bilder/specials/marodes/ordnung-am-arbeitsplatz-f43b3a46-27b5-4eee-8028-93e9d149d833.jpg

http://core0.staticworld.net/images/idge/imported/imageapi/2014/09/11/16/slide_image_01-09ss-java-hate-100419404-gallery.idge.jpg

http://www.laviva.com/uploads/pics/img_Keller_100-prozent_942px448px.jpg

http://www.hartung-trenz.de/projekte/2012/celle/celle1.jpg

http://cdn.osxdaily.com/wp-content/uploads/2010/02/macbook-pro-with-portrait-mode-display.jpeg