



## Inhaltsverzeichnis

<b>1</b>	<b>Herzlich Willkommen! .....</b>	<b>2</b>
<b>2</b>	<b>Das Projekt .....</b>	<b>3</b>
<b>3</b>	<b>Was wird benötigt .....</b>	<b>4</b>
<b>4</b>	<b>Programmieren .....</b>	<b>5</b>
4.1	Das Pinguin Projekt .....	5
4.1.1	Bilddateien herunterladen! .....	5
4.1.2	Step 1: Projektstruktur erstellen .....	5
4.1.3	Step 2: Mache <div>s .....	6
4.1.4	Step 3: Rollover Effekt .....	8
4.1.5	Step 4: Zeige die Pinguine an! .....	9
4.1.6	Step 5: Yeti alert! .....	10
4.1.7	Step 6: <i>Mix it up and challenge someone</i> .....	11
<b>5</b>	<b>Weitere Infos .....</b>	<b>12</b>
5.1	Links .....	12



## 1 Herzlich Willkommen!

Zuerst einmal: wir finden es toll, dass du dich für eine Lehre als InformatikerIn interessierst! Wir benötigen gut ausgebildeten und hoch motivierten Nachwuchs. Unser Job ist mega-spannend und vielseitig und wir hoffen dir in den 2 Tagen Schnupperlehre bei uns etwas davon zeigen zu können.

Wir werden zusammen ein kleines Projekt durchziehen. Ziel soll es sein, dass du einerseits nach 2 Tagen einen Einblick in unseren Beruf hast, andererseits aber auch einen direkten Nutzen davon trägst. D.h. für dich sollten danach Begriffe wie Betriebssystem, Shell Environment, Web, HTML, CSS, Java-Script etc. keine Fremdwörter mehr sein. Dann wären wir ziemlich

stolz auf uns, hätten wir unser Ziel doch voll erreicht.....



## 2 Das Projekt

Unser kleines Projekt beinhaltet folgendes

- Erstellen einer simplen Webseite

Das tönt jetzt nach wenig und ziemlich einfach. Uns geht's aber nicht darum, dass du eine möglichst komplizierte und aufwändige Webseite erstellst, sondern dass du in erster Linie verstehst, wie Webtechnologie funktioniert. Heisst, du verstehst, was du machst. In unserem Beruf ist dies etwas vom Wichtigsten...



### **3 Was wird benötigt**

Für das Projekt werden verschiedene Dinge benötigt: Hardware und Software

Hier ist eine Liste der benötigten Dinge:

#### **Hardware**

- Laptop

#### **Software**

- Web Browser, Firefox
- Visual Studio, Code-Editor

## 4 Programmieren

Wir kommen jetzt zum Programmieren. Wir verwenden verschiedene Web-Technologien, um die Seiten zu erstellen. Primär ist dies HTML, CSS und JavaScript.

Mit einem Code-Editor erstellen wir die Files und rufen diese im Webbrowser lokal auf.

### 4.1 Das Pinguin Projekt

Als Beispiel machen wir das Pinguin Projekt (im Internet zu finden unter: [http://googlecreativelab.github.io/coder-projects/projects/pop\\_up\\_penguins/](http://googlecreativelab.github.io/coder-projects/projects/pop_up_penguins/)). Nachfolgend eine deutsche Übersetzung der englischen Anleitung. Du darfst natürlich auch die englische Anleitung benutzen, falls dir das lieber ist. Diese findest du in dem oben angegebenen Link.

#### 4.1.1 Bilddateien herunterladen!

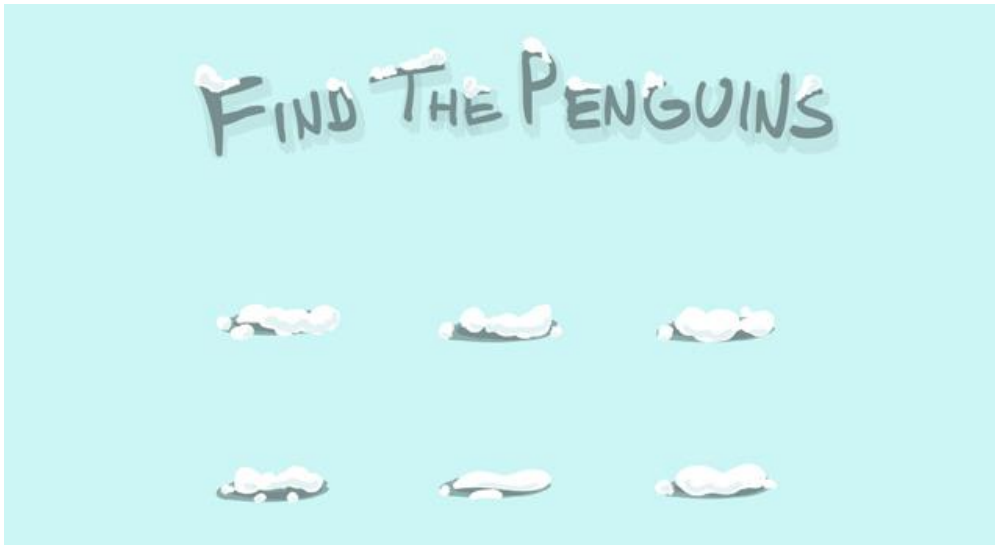
Bevor wir anfangen können, brauchen wir die Bilder der Pinguine: [penguin pngs](#)

#### 4.1.2 Step 1: Projektstruktur erstellen

Als erstes werden wir die Projektstruktur erstellen. Dazu erstellen wir einen neuen Ordner und benennen diesen "PinguinProjekt". In diesem Ordner erstellen wir weitere Ordner, einen namens "css" für das Stylesheet, die Darstellung, einen weiteren namens "js" für das JavaScript und den letzten namens "media" für die Bilder.

Da wir die Bilder schon heruntergeladen haben, können wir sie in den Ordner "media" entpacken.

### 4.1.3 Step 2: Mache <div>s



Jetzt, wo wir alles nötige haben um zu starten, können wir mit dem Coden beginnen.

- Öffne den Code-Editor
- Erstelle ein neues File namens "index.html" und speichere es in dem Ordner "PinguinProjekt"
- Schreibe darin den folgenden Code

```
<!DOCTYPE html>
<html>

<head>
  <title>Penguin Pop-Up</title>
  <meta charset="utf-8">
  <link href="css/index.css" media="screen" rel="stylesheet"
type="text/css"/>
  <script type="text/javascript" src="js/jquery.min.js"></script>
  <script src="js/index.js"></script>
</head>

<body>
  <div id="gameholder">
    <div id="title"></div>
    <div class="penguin1"></div>
    <div class="penguin2"></div>
    <div class="penguin3"></div>
    <div class="penguin4"></div>
    <div class="penguin5"></div>
    <div class="penguin6"></div>
    <div class="penguin7"></div>
    <div class="penguin8"></div>
    <div class="yeti"></div>
  </div>
</body>

</html>
```

Was macht dieser Code?



Diese divs definieren die Struktur des Spiels. In jedem "**penguin**" div wird sich einmal ein Pinguin oder ein Yeti verstecken. Die Bilder werden mit Hilfe von CSS hinzugefügt.

- Als Nächstes, erstelle nochmals ein neuen File namens "index.css" in dem Ordner "css"
- Füge den folgenden Inhalt hinzu:

```
body {
    background-color: #ccf5f5;
}

#gameholder {
    width: 600px;
    margin-left: auto;
    margin-right: auto;
}

#title {
    width: 600px;
    height: 150px;
    background-image:url('../media/penguin_title.png');
}
```

*Was macht dieser Code?*

- Im **body** wird die blaue Hintergrundfarbe definiert.
- **#gameholder** macht das **gameholder** div **600px** (Pixel) breit und zentriert es auf dem Bildschirm. Man sieht es nicht, aber es hilft uns, die anderen Elemente zu organisieren und positionieren.
- **#title** setzt die Grösse des **title** div und gibt ihm ein Hintergrundbild. Die URL zum Bild, ist der Pfad ab dem aktuellen Standort

Jetzt haben wir den Titel. Als nächstes machen wir kleine Schneehügel. Füge folgenden Code dem CSS File hinzu:

```
.penguin1 {
    width: 200px;
    height: 200px;
    float: left;
    background-image:url('../media/mound_1.png');
}
```

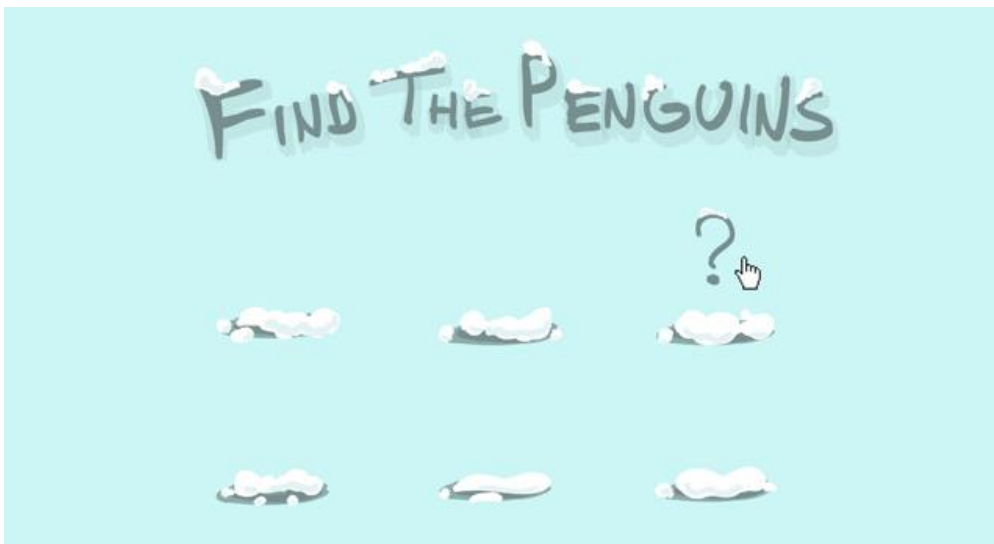
Das gesamte CSS sieht jetzt so aus:



```
body {  
    background-color: #ccf5f5;  
}  
  
#gameholder {  
    width: 600px;  
    margin-left: auto;  
    margin-right: auto;  
}  
  
#title {  
    width: 600px;  
    height: 150px;  
    background-image: url('../media/penguin_title.png');  
}  
  
.penguin1 {  
    width: 200px;  
    height: 200px;  
    float: left;  
    background-image: url('../media/mound_1.png');  
}
```

Mach jetzt noch mehr Penguin Klassen und die Yeti Klasse. Kopiere die **.penguin1** Klasse und nenne sie **.penguin2**, **.penguin3** etc. Für die Yeti Klasse kannst du auch die **.penguin1** Klasse kopieren und nennst sie **.yeti**.

#### 4.1.4 Step 3: Rollover Effekt



Für den Rollover Effekt werden wir die **Pseudoklasse :hover** verwenden. Bei jedem Rollover sollte über dem Schneehügel ein Fragezeichen stehen. Im Gegensatz zu normalen Klassen, werden Pseudoklassen vom Browser selbst eingefügt. Wenn man mit der Maus über ein Objekt fährt, bekommt dieses die Pseudoklasse **:hover**.



Das merkt auch das CSS und zeigt die entsprechend definierten Styles an. Es gibt mehrere Pseudoklassen, z.B. **:active** für einen Mausklick, **:focus** für ein Textfeld welches gerade bearbeitet wird etc. Konkret sieht das so aus:

- Füge unter jedem **.penguin** folgenden Code hinzu:

```
.penguin1:hover {  
    background-image:url('../media/mound_1_hover.png');  
    cursor: pointer;  
}
```

So wie in diesem Beispiel, wird eine Pseudoklasse angewendet. Die bisherigen Einstellungen werden überschrieben, wenn sie ausgeführt wird. Es ist eine Möglichkeit, Effekte hinzuzufügen. In diesem Fall werden wir das Hintergrundbild ändern, wenn man mit der Maus über das div fährt. Es wird auch der Mauszeiger geändert; neu wird ein Pointer (kleine Hand) angezeigt.

- Achte darauf, dass du die Nummern bei dem Klassen-Namen und bei dem Bild anpasst.

Nun kannst du dies anschauen, indem du das File im Browser aufrufst. Gebe den vollständigen Pfad des "index.html" Files in der Suchleiste an.

#### 4.1.5 Step 4: Zeige die Pinguine an!



Wie wir gesehen haben, wird die pseudoklasse **:hover** aufgerufen, wenn man mit der Maus darüber fährt. Es gibt auch eine Pseudoklasse welche aufgerufen wird, wenn man auf ein Element klickt: **:active**.

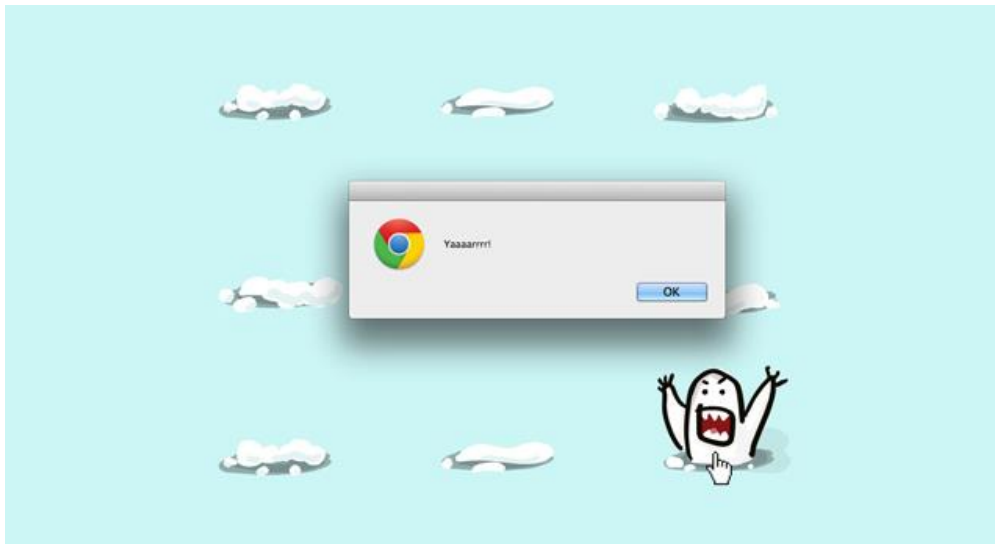
- Füge unter **.penguin1:hover** folgenden Code an:

```
.penguin1:active {  
    background-image:url('../media/penguin_1.png');  
}
```

Die Position der Pseudoklasse ist wichtig. Die **:active** Klasse sollte immer unter die **:hover** Klasse kommen, sodass **:hover** nicht **:active** überschreibt.

- Öffne das File im Browser und klicke auf den ersten Schneehügel. Wurde ein Pinguin angezeigt? Halte die Maustaste gedrückt um den Pinguin nicht wieder verschwinden zu lassen.
- Führe diesen Schritt für jeden Pinguin und den Yeti durch.

#### 4.1.6 Step 5: Yeti alert!



Wir haben alle Pinguine und den Yeti an ihren Platz gesetzt. Machen wir jetzt ein Popup, wenn man auf den Yeti klickt.

- Erstelle ein neues File namens "index.js" im Ordner "js".
- Schreibe folgenden Code hin.

```
$(document).ready( function() {  
    //This code will run after your page loads  
    $(".yeti").mouseup(function() {  
        alert("Yaaaarrrr!");  
    });  
});
```

Das ist ein bisschen JavaScript mit jQuery, welches dafür sorgt, dass ein Popup mit dem Text "Yaaaarrrr!" erscheint. Probiere es aus!

#### 4.1.7 Step 6: *Mix it up and challenge someone*



Gratulation, du kannst jetzt anfangen zu spielen! Mixe die Klassen im HTML, um den Yeti an eine andere Position zu verschieben.

## 5 Weitere Infos

### 5.1 Links

Link	Beschreibung
<a href="http://www.w3schools.com">http://www.w3schools.com</a>	W3 Schools ist eine gute Referenz für Webtechnologien. Man findet Informationen und Tutorials zu HTML, CSS, Javascript und noch vielen weiteren Sprachen. (Die Seite ist auf Englisch, also werden gewisse Sprachkenntnisse vorausgesetzt)
<a href="http://wiki.selfhtml.org/wiki/Startseite">http://wiki.selfhtml.org/wiki/Startseite</a>	SelfHTML ist die beste deutsche HTML Referenz welche man im Internet finden kann. Man findet Informationen zu HTML, CSS, Javascript. Im Gegensatz zu W3Schools gibt es jedoch keine Tutorials.