Breaking the Code to Inclusion

Designing Micro Materials Based on PRIMM Principles for Accessible Programming Education
Introduction
Background
```java
package com.example.demo;

public class Main {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```
PRIMM

Not Mine
Predict
Run
Investigate
Modify
Make

Partly Mine

All Mine

- Sentance et al., 2019
Micromaterials

- Open Education Resources
- Should be easy to integrate in existing curricula
- Should provide automated feedback
- Ideally content should be generated automatically

- Adam Leskis., 2008
HTML StuddyBuddy
Design Selection
Select the design you want to implement.

Titles and Paragraphs

- Welcome Party
- Employee Manual
- About WISE

Lists

- Shopping List
- Muffin Recipe

Adding Exercises/Flashing Tutorial
EXAMPLE

<h1> This is my main heading </h1>
<h2> This is a subheading </h2>
<h3> This is a subsubheading </h3>
Welcome Party

Welcome party

No actual design yet. But we have a few cool paragraphs here, try to replicate them.


Welcome Party

Hints

This is the very first design that should get you used to using headings and paragraphs.

Here are some elements you will probably need:

<p> element

The <p> HTML element represents a paragraph. Paragraphs are usually represented in visual media as blocks of text separated from adjacent blocks by blank lines and/or first-line indentation, but HTML paragraphs can be any structural grouping of related content, such as images or form fields.

Read more on the documentation page

<h1> element

The <h1> to <h6> HTML elements represent six levels of section headings. <h1> is the highest section level and <h6> is the lowest.

Read more on the documentation page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam at pretium libero. Elit vitae magna dignissim mauris sit amet lorem a mauris elit.
SQL StuddyBuddy
SQL StudyBuddy
An easy way to practise SQL

- DATA QUERY LANGUAGE
- DATA MANIPULATION LANGUAGE
- DATA DEFINITION LANGUAGE
The WISE Wishlist database has only one table, **items**.

In this items table the data is stored for all items that WISE lab members might want to buy in the near future for their research purposes. Every entry in the table contains an id for the item, the name of the item, the price of the item and the first name of the WISE member that wants to buy the item.
What is the price of the oscilloscope (use . for decimals)?

```
select * from items
```

Answer: 289.99

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>price</th>
<th>member</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Oscilloscope</td>
<td>289.99</td>
<td>Maxim</td>
</tr>
<tr>
<td>7</td>
<td>Ipad Pro 11 Inch</td>
<td>1149</td>
<td>Beat</td>
</tr>
<tr>
<td>8</td>
<td>Photocube</td>
<td>129</td>
<td>Beat</td>
</tr>
<tr>
<td>9</td>
<td>LEDs</td>
<td>0.2</td>
<td>Maxim</td>
</tr>
<tr>
<td>10</td>
<td>Voltage meter</td>
<td>40</td>
<td>Maxim</td>
</tr>
</tbody>
</table>
```
insert into items (name, price, member) VALUES
("Macbook Pro 32GB", 2599, "Yoshi");
```

**Question**

WISE member Yoshi would like to add the item ‘Macbook Pro 32GB’ to the wishlist, the model costs 2599€.

The WISE Wishlist database has only one table, `items`. In this table the data is stored for all items that WISE lab members might want to buy in the near future for their research purposes. Every entry in the table contains an id for the item, the name of the item, the price of the item and the first name of the WISE member that wants to buy the item.

**Checks**

- The table should have 19 elements
- The WISE member should be Yoshi
- The item name should be Macbook Pro 32GB
- The price should be 2599€
CREATE TABLE ticket_sales (
  "id" INTEGER NOT NULL,
  "amount" INTEGER NOT NULL UNIQUE,
  "type" TEXT NOT NULL,
  "price" REAL NOT NULL,
  PRIMARY KEY("id" AUTOINCREMENT)
);
Code Tracer

Select a trace type:

Beginner
Variable Steps
Keep track of the steps that all the variables go through throughout the program. Perfect for beginners.

OPEN THE TABLE

Beginner
Operators Table
Basic trace table to log the use of the operators throughout the program’s runtime.

OPEN THE TABLE

Intermediate
Variable Values
Keep track of the values of all the variables throughout the program, a little more advanced.

OPEN THE TABLE

Advanced
Current Values
Keep track of the current values in the program, recommended for advanced users.

OPEN THE TABLE
<table>
<thead>
<tr>
<th>line</th>
<th>name</th>
<th>action</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>x</td>
<td>declare, init</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>declare, init</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>x</td>
<td>read</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>read</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>area</td>
<td>declare, init</td>
<td>375</td>
</tr>
<tr>
<td>Expression</td>
<td>Evaluates to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>! isOk</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x ++</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x - y</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>isOk ? &quot;yes&quot; : &quot;no&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Evaluates to</th>
</tr>
</thead>
<tbody>
<tr>
<td>! isOk</td>
<td>true</td>
</tr>
<tr>
<td>x ++</td>
<td>8</td>
</tr>
<tr>
<td>x - y</td>
<td>5</td>
</tr>
<tr>
<td>isOk ? &quot;yes&quot; : &quot;no&quot;</td>
<td></td>
</tr>
<tr>
<td>Line number</td>
<td>Value</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>375</td>
</tr>
<tr>
<td>variable</td>
<td>value</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>x</td>
<td>15</td>
</tr>
<tr>
<td>y</td>
<td>25</td>
</tr>
<tr>
<td>isOk</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>Yoshi</td>
</tr>
<tr>
<td>area</td>
<td>375</td>
</tr>
</tbody>
</table>
3 Types of Cards

Goal Cards

assert(y < z &&
z < x,
"happy days :)"
);

Environment Cards

let foo = 65;
let bar = 90;
let x = 43;
let y = 31;
let z = 28;

if (z >= bar) {
  z = z + 23;
  bar = bar - 96;
}

Code Cards
1–4 Players

Add players

Player name

Add player

Yoshi

Evan
The Playing Field
Playing a Card
Updating the Trace Table
Earning Points
KING'S SCROLL
The Search for the Chosen One
```javascript
let helmet = true;
let shield = true;
let sword = true;
let cape = true;
let z = 12;
if (z > 0) {
  helmet = shield
}
```
Sixteen Heroes

- Shield
- Sword
- Helmet
- Cape
- Gender
- Skin tone
let helmet = true;
let shield = true;
let sword = false;
let cape = false;
let h = shield;
shield = cape;
cape = h;
let k = 6;
if (k < 13) {
    helmet = sword;
} else {
    shield = true;
}

let helmet = true;
let sword = true;
let shield = false;
let cape = false;
let p = cape;
cape = helmet;
helmet = p;
let q = 7;
do {
    q++;
    cape = !sword;
} while (q <= 18);

let cape = true;
let helmet = false;
let shield = false;
let sword = false;
function 1O {
    sword = !cape;
}
cape = cape;
1O;
let m = cape;
cape = shield;
shield = m;
let helmet = true;
let shield = false;
let sword = false;
let cape = false;
for (let d = 0; d < 12; d++) {
    cape = true;
}
Settings

Difficulty
- Easy
- Medium
- Hard

Language features
- for loops
- if
- if ... else ...
- do while loops
- function declarations

Back to main menu
## State Table

<table>
<thead>
<tr>
<th>Xuyao</th>
<th>Arun</th>
<th>Elena</th>
<th>Gelila</th>
<th>Eleni</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Character 1" /></td>
<td><img src="image2.png" alt="Character 2" /></td>
<td><img src="image3.png" alt="Character 3" /></td>
<td><img src="image4.png" alt="Character 4" /></td>
<td><img src="image5.png" alt="Character 5" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Piet</th>
<th>Eilen</th>
<th>Christophe</th>
<th>Renny</th>
<th>Migueley</th>
<th>Isaac</th>
<th>Maxim</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image6.png" alt="Character 6" /></td>
<td><img src="image7.png" alt="Character 7" /></td>
<td><img src="image8.png" alt="Character 8" /></td>
<td><img src="image9.png" alt="Character 9" /></td>
<td><img src="image10.png" alt="Character 10" /></td>
<td><img src="image11.png" alt="Character 11" /></td>
<td><img src="image12.png" alt="Character 12" /></td>
</tr>
</tbody>
</table>

### Code Snippet
```javascript
let helmet = true;
let shield = true;
let sword = true;
let cape = true;
let z = 12;
if (z == 0) {
  helmet = shield;
}
```

### State Controls
- [ ] shield
- [ ] sword
- [ ] cape
- [ ] helmet

### Count
2
Guidelines
Embrace themes!
But don’t forget about the skill transfer principle
Invite the social aspects!
Keep the setup minimal
Focus on one specific learning goal

<table>
<thead>
<tr>
<th>Specific</th>
<th>Clear and specific outcomes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurable</td>
<td>Define assessment/evaluation to measure outcomes?</td>
</tr>
<tr>
<td>Achievable</td>
<td>Is the expected level realistic?</td>
</tr>
<tr>
<td>Relevant</td>
<td>Is the goal relevant for the general goals of the learners?</td>
</tr>
<tr>
<td>Time-limited</td>
<td>Can it be achieved in a logical time unit?</td>
</tr>
</tbody>
</table>
Keep in mind the expertise reversal principle
Automated content generation will be a lifesaver
Make your things mobile compatible
Join us and build things

Raspberry Pi

CoderDojo

Hack Your Future

Migra Code