




Yoshi Malaise

ymalaise@vub.be

 0000-0002-3228-6790



FOSDEM 2023

Breaking the Code to Inclusion

Designing Micro Materials Based on PRIMM Principles for Accessible
Programming Education



VRIJE
UNIVERSITEIT
BRUSSEL



WEB & INFORMATION
SYSTEMS ENGINEERING

Introduction





HUB

HACK
YOUR
FUTURE

Background

Project Structure:

- demo (~/Downloads/ymlaise-thesis-2020-master/code)
 - .idea
 - .mvn
 - src
 - main
 - java
 - com.example.demo
 - HelloApplication
 - HelloController
 - Main
 - module-info.java
 - resources
 - target
 - .gitignore
 - demo.iml
 - mvnw
 - mvnw.cmd
 - pom.xml
 - External Libraries
 - Scratches and Consoles

```
1 package com.example.demo;
2
3 public class Main {
4     public static void main(String[] args) {
5         System.out.println("Hello world!");
6     }
7 }
8
```

Notifications

Maven

Build: Sync x Build Output x

demo: build failed At 22/11/2022, 10:15 with 1 error 1 sec, 219 ms

- Main.java src/main/java/com/example/demo 1 error
 - ';' expected :5

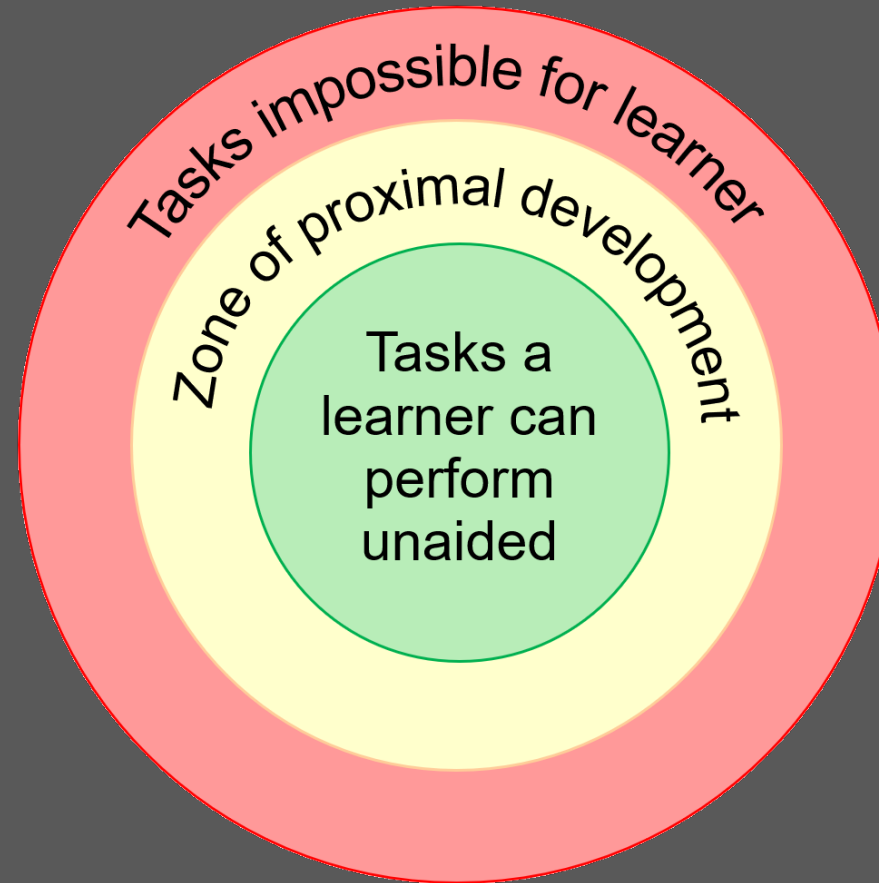
[/Users/yoshimalaise/Downloads/ymlaise-thesis-2020-master/code/rs1-backend/demo/src/main/java/com/example/demo/Mai](#)
java: ';' expected



PRIMM

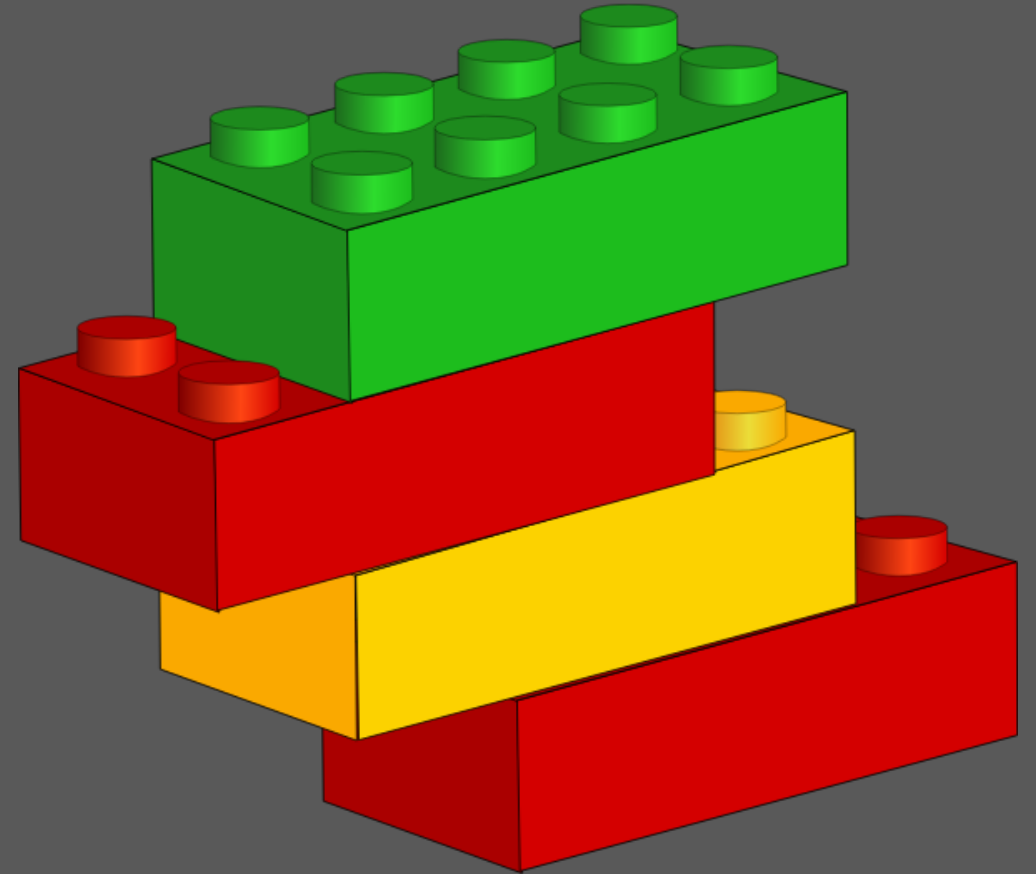


Zone of Proximal Development



Micromaterials

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



Examples

HTML StuddyBuddy

← Level Select

Design Selection

Select the design you want to implement.

Titles and Paragraphs



Welcome Party

Employee Manual

About WISE

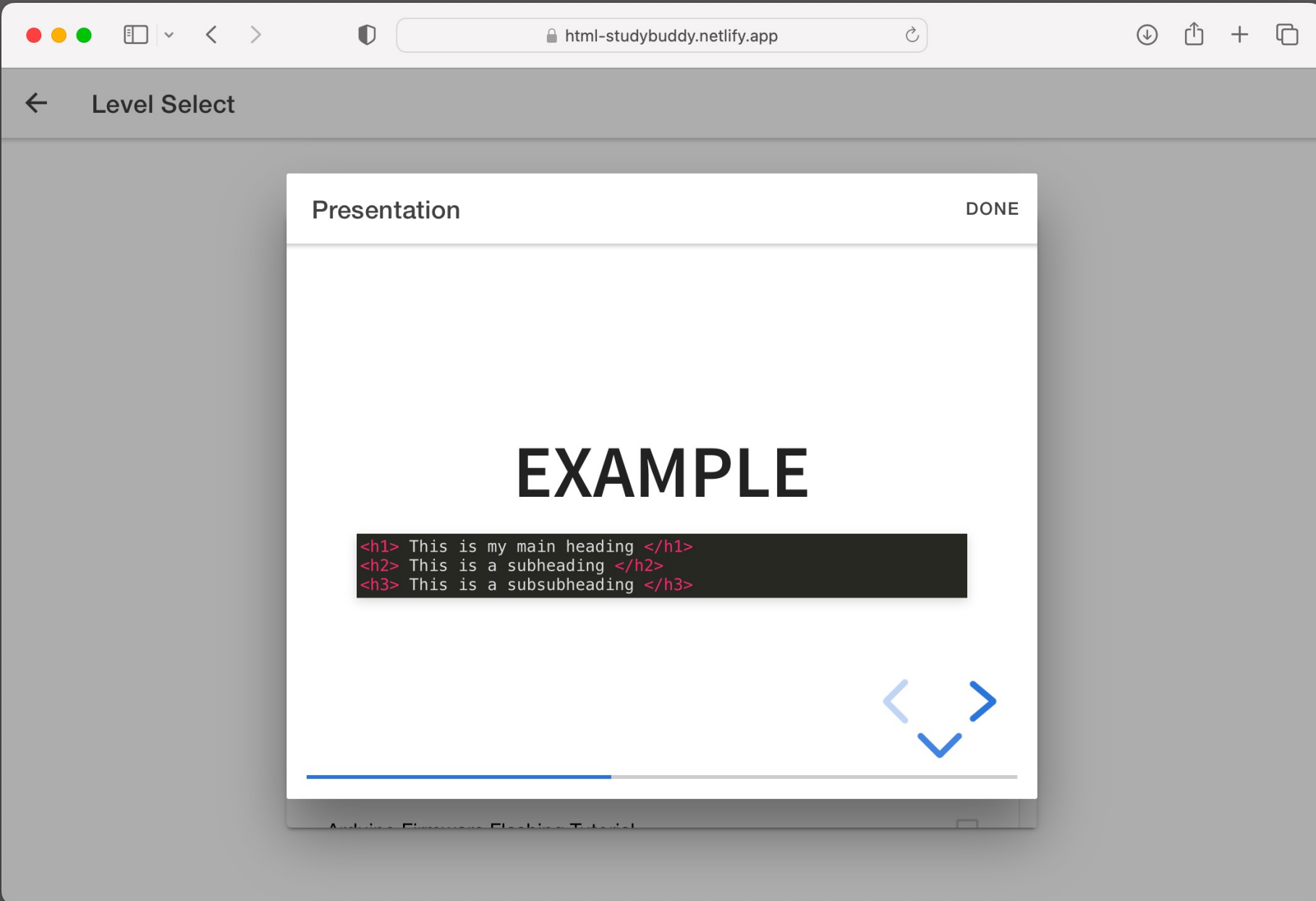
Lists

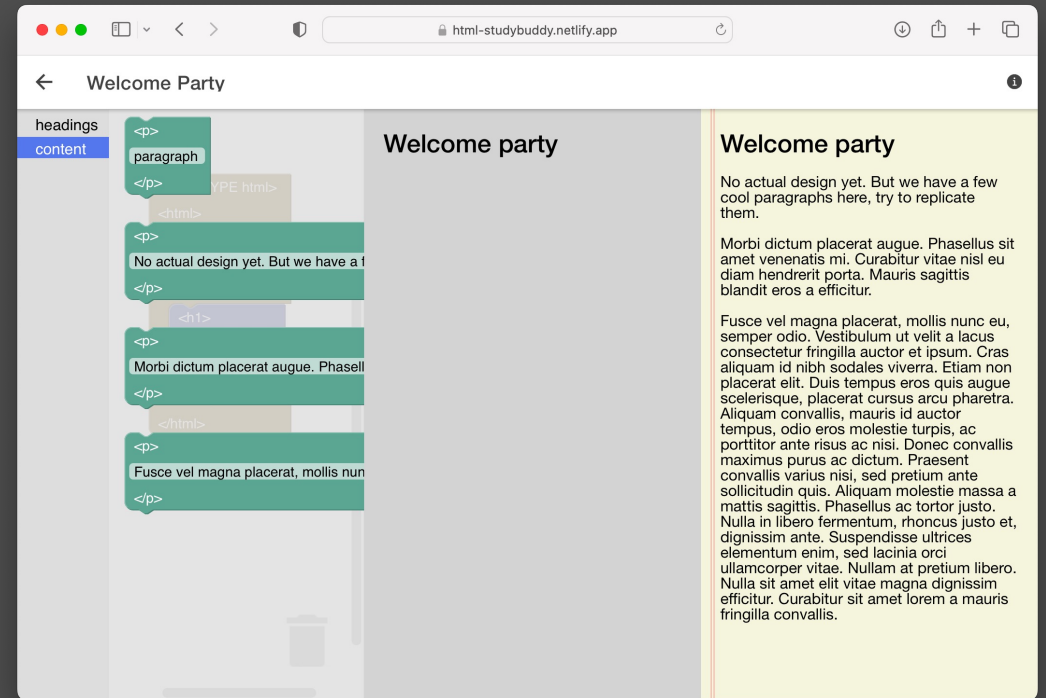
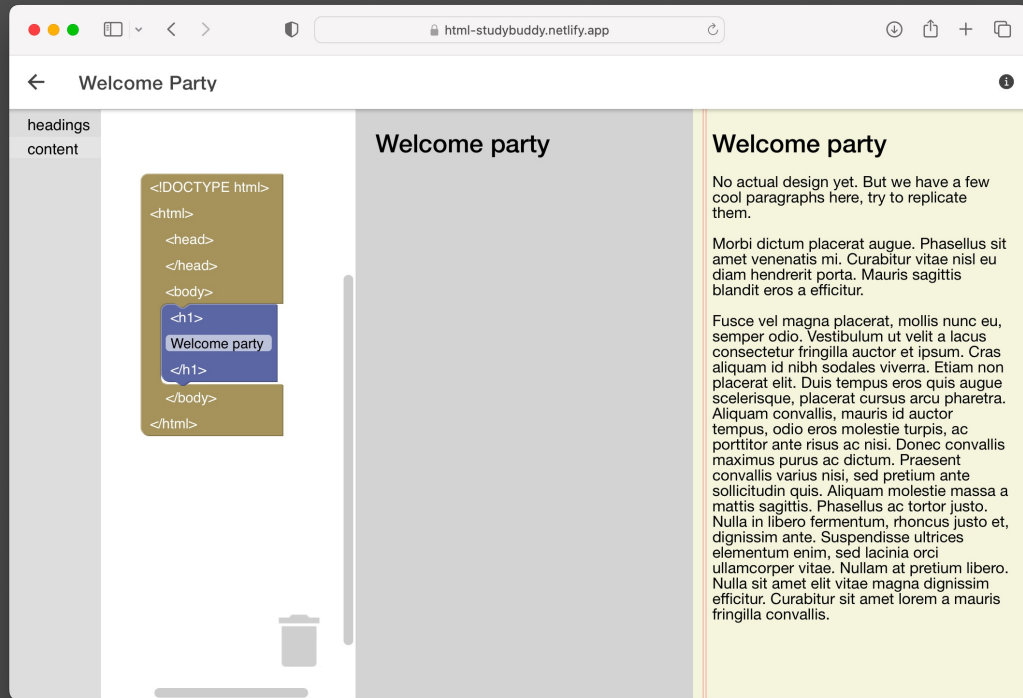


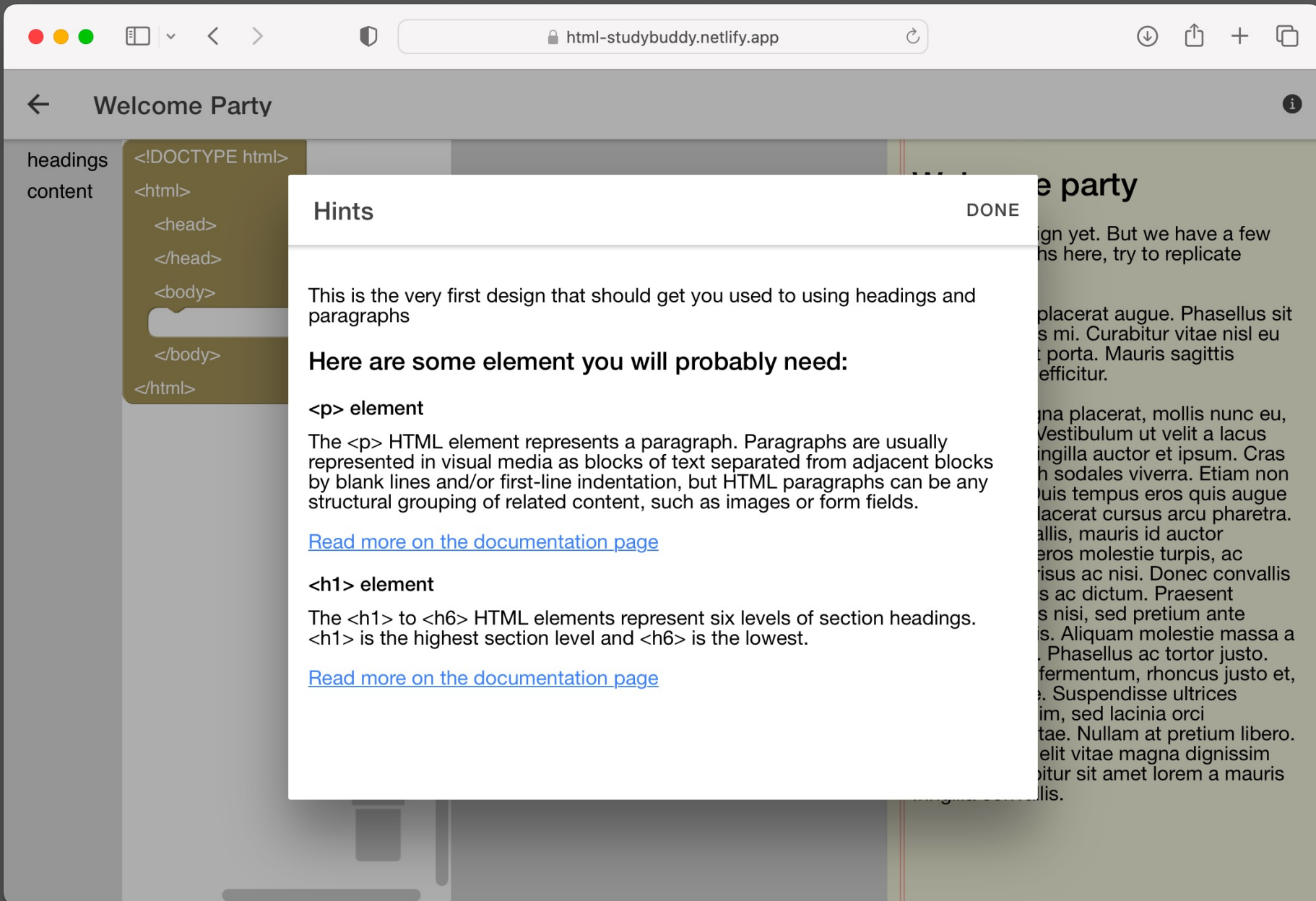
Shopping List

Muffin Recipe

And...







SQL StuddyBuddy



sql-studybuddy.netlify.app



SQL StudyBuddy

An easy way to practise SQL

DATA QUERY LANGUAGE

DATA MANIPULATION LANGUAGE

DATA DEFINITION LANGUAGE



sql-studybuddy.netlify.app



Code Editor

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40

Wise Wishlist DB DONE

items
id: number (PK)
name: varchar
price: float
member: varchar

The WISE Wishlist database has only 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.

32GB' to the wishlist, the model costs 2599\$.

at WISE lab members might want to buy
y entry in the table contains an id for
n and the first name of the WISE



Code Editor



```
1  
2 select * from items  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40
```

Question

What is the price of the oscilloscope (use . for decimals)?

Answer:



Query Result

Showing max 10 results

id	name	price	member
1	M5Stack	62.39	Yoshi
2	M5Stick C PLUS	29.8	Maxim
3	Quest Pro	1799.99	Maxim
4	Arduino Uno	24	Yoshi
5	Oculus Rift	400	Yoshi
6	Oscilloscope	299.99	Maxim
7	Ipad Pro 11 Inch	1149	Beat
8	Photocube	129	Beat
9	LEDs	0.2	Maxim
10	Voltage meter	40	Maxim



sql-studybuddy.netlify.app



Code Editor



```
1 insert into items (name, price, member) VALUES
2 ("Macbook Pro 32GB", 2399, "Yoshi");
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```

Question

WISE member Yoshi would like to add the item 'Macbook Pro 32GB' to the wishlist, the model costs 2599\$.

items
id: number (PK) name: varchar price: float member: varchar

The WISE Wishlist database has only 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.

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

Code Editor

```
1 CREATE TABLE ticket_sales (  
2     "id" INTEGER NOT NULL,  
3     "amount" INTEGER NOT NULL UNIQUE,  
4     "type" TEXT NOT NULL,  
5     "price" REAL NOT NULL,  
6     PRIMARY KEY("id" AUTOINCREMENT)  
7 );  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40
```

Question

Create the following db

ticket_sales

- id: integer PK auto-increment
- amount: integer
- type: text
- price: real

Checks

table ticket_sales exists



could insert two records



auto increment is working



TraceTables

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](#)

trace-tables.netlify.app

← Variable steps

	line	name	action	value
<input type="checkbox"/>	3	x	declare, init	15
<input type="checkbox"/>	4	y	declare, init	25
<input type="checkbox"/>	5	x	read	15
<input type="checkbox"/>	5	y	read	25
<input checked="" type="checkbox"/>	5	area	declare, init	375

ADD STEP

trace-tables.netlify.app

Operators Table

Expression	Evaluates to
! ▾ isOk	true
x ++ ▾	8
x - ▾ y	5
isOk ? "yes" : "no"	"no"

REMOVE ADD STEP

trace-tables.netlify.app

Operators Table

Expression	Evaluates to
! ▾ isOk	true
x ++ ▾	8
x - ▾ y	5
isOk ? "yes" : "no"	"no"

- Add Step
- Unary-Prefix
- Unary-Postfix
- Binary
- Ternary

trace-tables.netlify.app

Variable values

x

Line number	Value
3	15

REMOVE LINE ADD LINE

y

Line number	Value
4	25

REMOVE LINE ADD LINE

area









Line number	Value
5	375

REMOVE LINE ADD LINE

ADD VARIABLE

trace-tables.netlify.app

← Current values

variable	value	
x	15	 
y	25	 
isOk	<input checked="" type="checkbox"/>	
name	Yoshi	
area	375	 

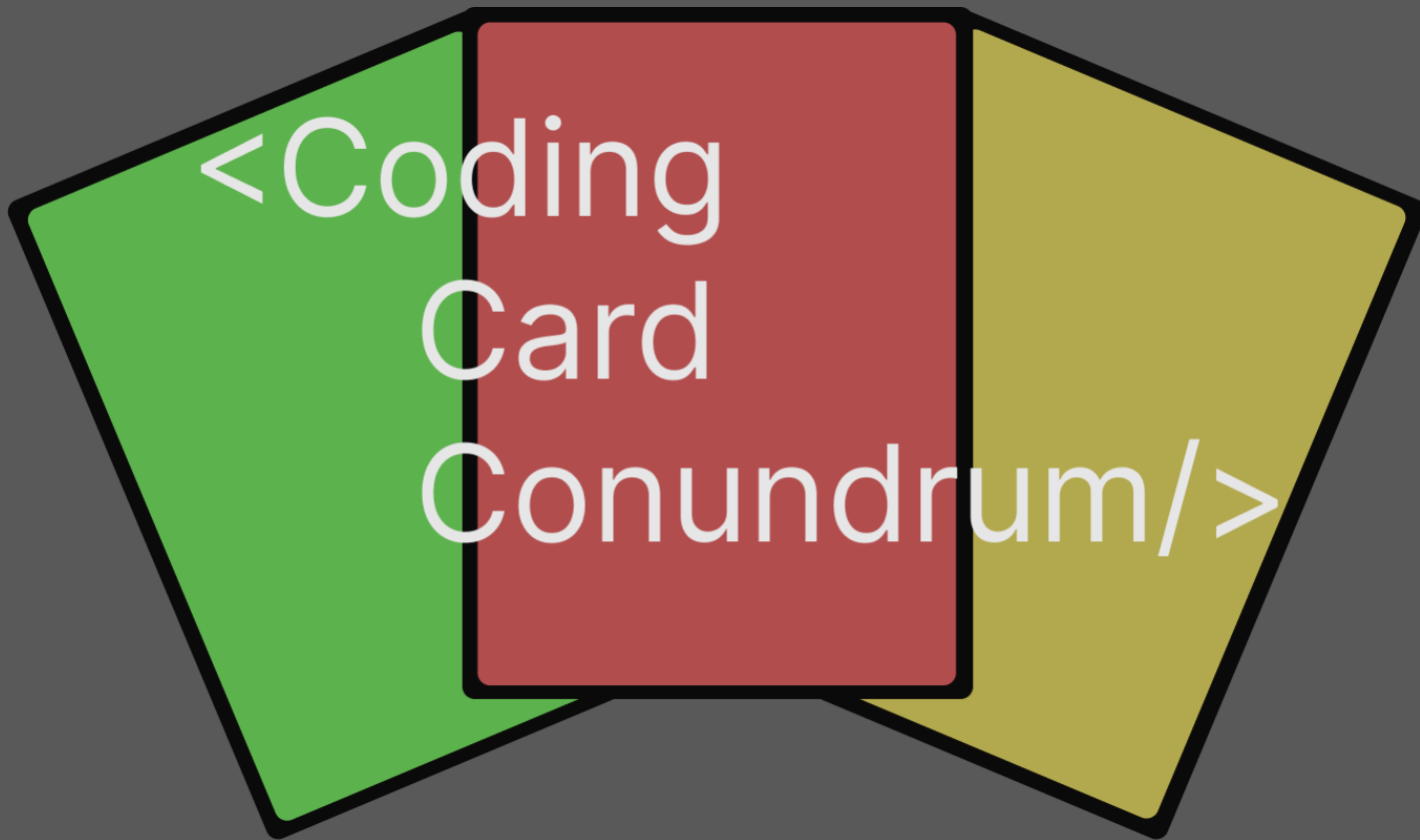
[ADD VARIABLE](#)

trace-tables.netlify.app

← Current values

variable	value	
x	15	↕ 🗑️
y	25	↕ 🗑️
isOk	<input checked="" type="checkbox"/>	🗑️
name	Yoshi	🗑️
area	375	↕ 🗑️

- Export trace
- Print to PDF
- Export as JSON



<Coding
Card
Conundrum/>

3 Types of Cards

Goal

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

- 2 -

Goal Cards

Environment

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

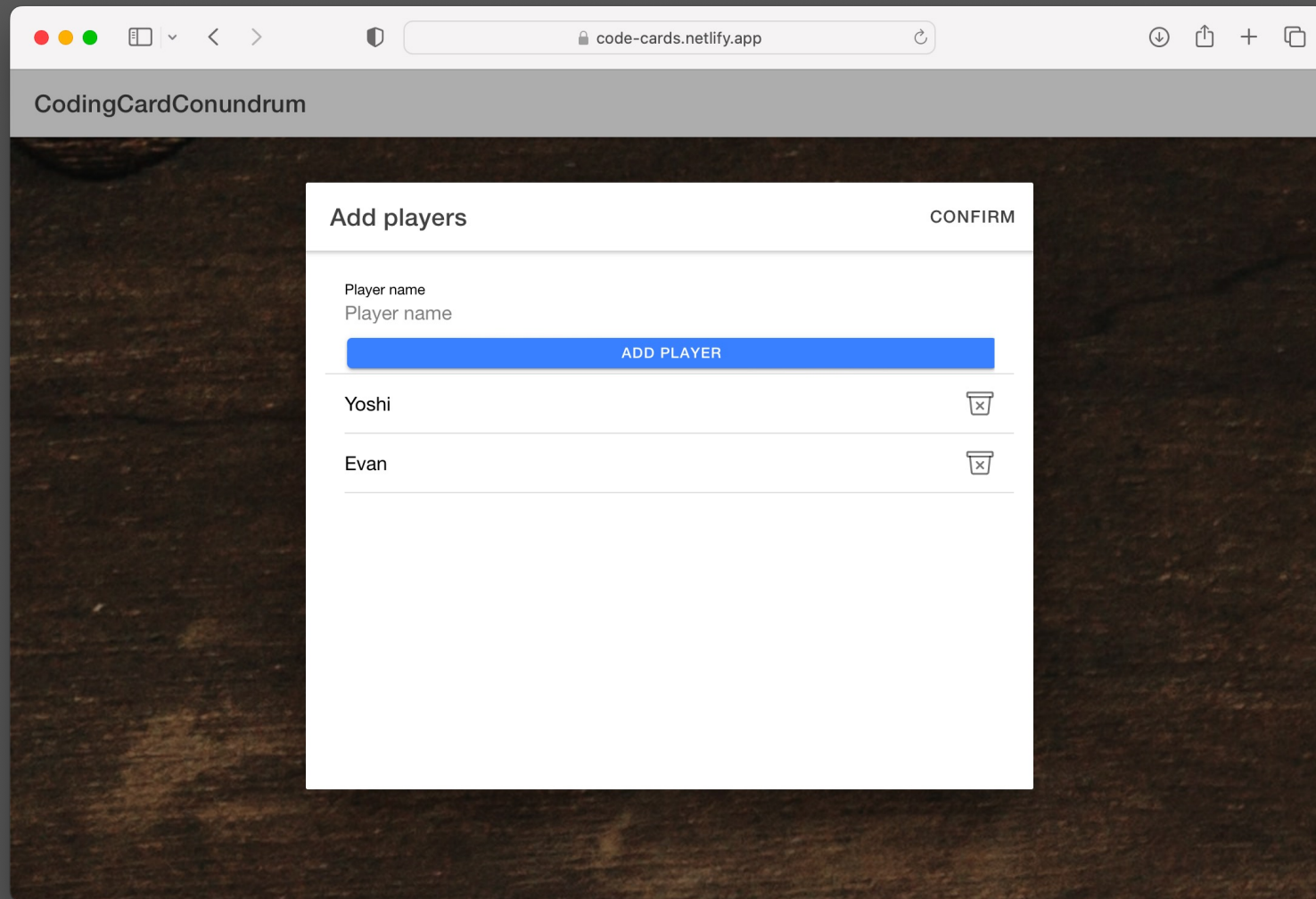
Environment Cards

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

**Heavy
mathematics**

Code Cards

1-4 Players



The Playing Field

code-cards.netlify.app

CodingCardConundrum - Yoshi's turn

foo	bar	x	y	z
6	74	38	16	63

Environment 1 -

```
let foo = 6;
let bar = 74;
let x = 38;
let y = 16;
let z = 63;
```

foo	bar	x	y	z
50	15	15	76	7

Environment 2 -

```
let foo = 50;
let bar = 15;
let x = 15;
let y = 76;
let z = 7;
```

foo	bar	x	y	z
51	61	44	44	75

Environment 3 -

```
let foo = 51;
let bar = 61;
let x = 44;
let y = 44;
let z = 75;
```

```
if (bar > foo) {
  x = bar;
  bar = foo;
  foo = x;
}
```

Swap if bigger

```
if (foo >= y) {
  foo = foo % 54;
  y = y + 36;
}
```

Heavy mathematics

```
if (x > z) {
  x = x % 58;
  z = z + 88;
}
```

Heavy mathematics

```
if (bar !== foo) {
  bar = bar % 12;
  foo = foo - 81;
}
```

Heavy mathematics

```
if (foo < bar) {
  y = foo;
  foo = bar;
  bar = y;
}
```

Swap if smaller

Goal

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

- 2 -

Playing a Card

code-cards.netlify.app

CodingCardConundrum - Yoshi's turn

foo	bar	x	y	z
6	74	38	16	63

Environment 1 -

```
let foo = 6;
let bar = 74;
let x = 38;
let y = 16;
let z = 63;
```

ADD CARD HERE

foo	bar	x	y	z
50	15	15	76	7

Environment 2 -

```
let foo = 50;
let bar = 15;
let x = 15;
let y = 76;
let z = 7;
```

ADD CARD HERE

foo	bar	x	y	z
51	61	44	44	75

Environment 3 -

```
let foo = 51;
let bar = 61;
let x = 44;
let y = 44;
let z = 75;
```

ADD CARD HERE

```
if (bar > foo) {
  x = bar;
  bar = foo;
  foo = x;
}
```

Swap if bigger

```
if (foo >= y) {
  foo = foo % 54;
  y = y + 36;
}
```

Heavy mathematics

```
if (x > z) {
  x = x % 58;
  z = z + 80;
}
```

Heavy mathematics

```
if (bar !== foo) {
  bar = bar % 12;
  foo = foo - 81;
}
```

Heavy mathematics

```
if (foo < bar) {
  y = foo;
  foo = bar;
  bar = y;
}
```

Swap if smaller

Goal

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

- 2 -

Updating the Trace Table

The screenshot shows a web browser window with the URL `code-cards.netlify.app`. The page title is "CodingCardConundrum - Yoshi's turn". The main content area is divided into three environments, each with a table of variables and a code editor.

Environment 1:

foo	bar	x	y	z
6	74	38	16	63

```
let foo = 6;
let bar = 74;
let x = 38;
let y = 16;
let z = 63;
```

Environment 3:

foo	bar	x	y	z
51	61	44	44	75

```
let foo = 51;
let bar = 61;
let x = 44;
let y = 44;
let z = 75;
```

Trace Table Dialog:

The dialog is titled "Trace Table" and has a "CONFIRM" button. It contains a code editor with the following code:

```
let foo = 50;
let bar = 15;
let x = 15;
let y = 76;
let z = 7;

if (bar !== foo) {
  bar = bar % 12;
  foo = foo - 81;
}
```

Below the code editor is a table with the following data:

#	foo	bar	x	y	z
1	50				
2		15			
3			15		
4				76	
5					7

There is an "ADD LINE" button above the table.

Earning Points

The screenshot shows a web browser window with the URL `code-cards.netlify.app`. The page title is "CodingCardConundrum - Yoshi's turn". The interface features three environments and a central goal modal.

Environment 1:

foo	bar	x	y	z
6	74	38	16	63

```
let foo = 6;  
let bar = 74;  
let x = 38;  
let y = 16;  
let z = 63;
```

Environment 3:

foo	bar	x	y	z
51	61	44	44	75

```
let foo = 51;  
let bar = 61;  
let x = 44;  
let y = 44;  
let z = 75;
```

Goal Reached! Modal:

Yoshi

CONFIRM

Goal

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

- 2 -

+ 2

WING'S SCROLL

The Search for the Chosen One





kings-scroll.netlify.app



Back to homescreen

```

let helmet = true;
let shield = true;
let sword = true;
let cape = true;
let z = 12;
if (z >= 0) {
  helmet = shield;
}

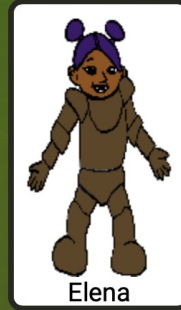
```



Xuyao



Arun



Elena



Gelila



Eleni



Piet



Elien



Christophe



Renny



Migdeily



Isaac



Maxim



Olga



Yoshi



Beat



Ekene

shield sword cape helmet

Sixteen Heroes

- Shield
- Sword
- Helmet
- Cape
- Gender
- Skin tone

The screenshot shows a web browser window with the URL `kings-scroll.netlify.app`. The page features a dark green background. On the left, there is a scrollable code editor with the following code:

```
let helmet = true;
let shield = true;
let sword = true;
let cape = true;
let z = 12;
if (z >= 0) {
  helmet = shield;
}
```

Below the code editor are four checkboxes: `shield`, `sword`, `cape`, and `helmet`, all of which are currently unchecked. A text input field labeled `z` is positioned below the checkboxes.

On the right side of the page, there is a grid of 16 hero cards, each with a unique character illustration and a name below it:

- Xuyao
- Arun
- Elena
- Gelila
- Eleni
- Piet
- Elien
- Christophe
- Renny
- Migdeily
- Isaac
- Maxim
- Olga
- Yoshi
- Beat
- Ekene

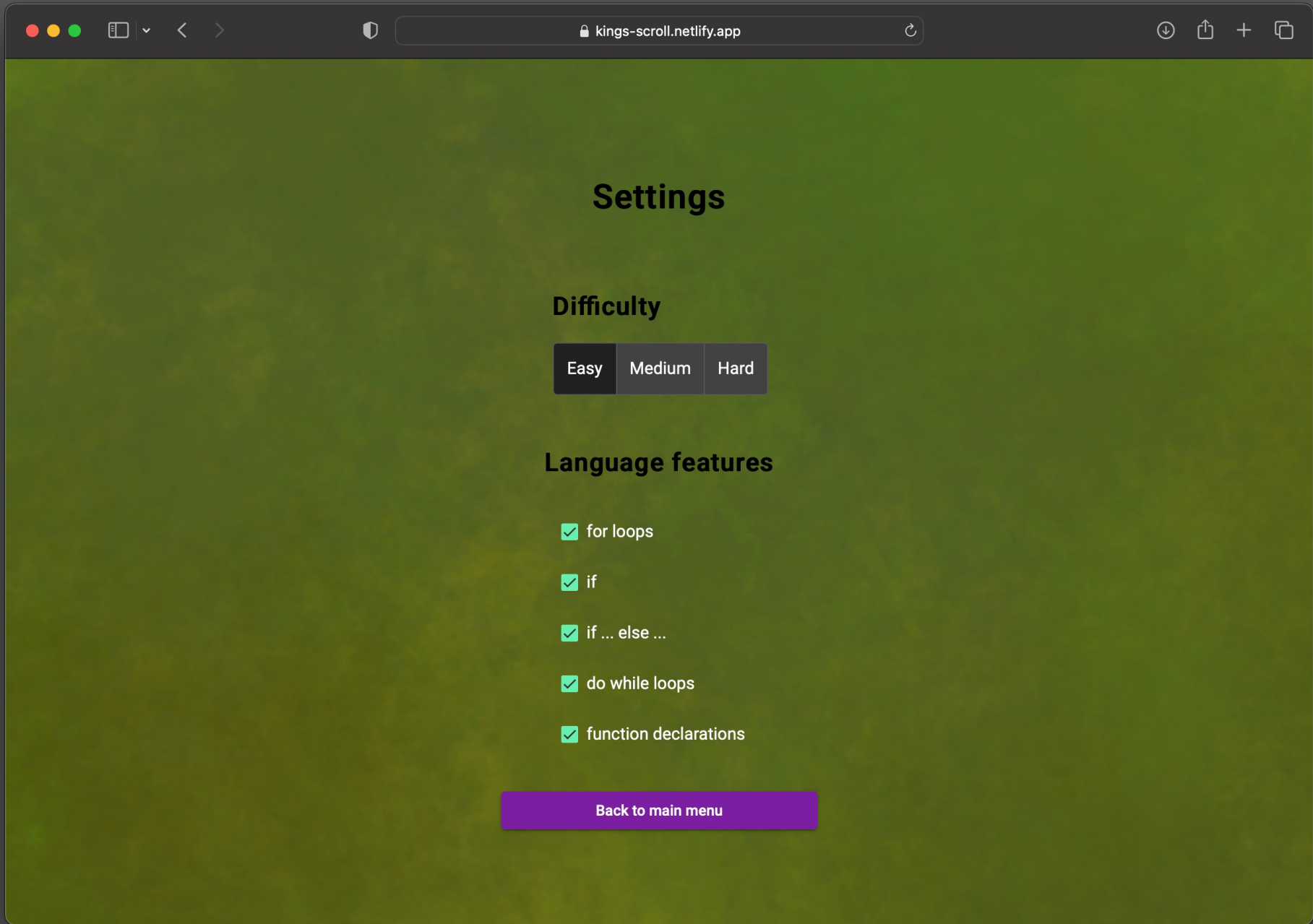
The Scroll

```
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 l() {
  sword = !cape;
}
cape = cape;
l();
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

Back to homescreen

```
let helmet = true;
let shield = true;
let sword = true;
let cape = true;
let z = 12;
if (z >= 0) {
  helmet = shield;
}
```

Character cards (names):

- Xuyao, Arun, Elena, Gelila, Eleni, Piet
- Elien, Christophe, Renny, Migdeily, Isaac, Maxim
- Olga, Yoshi, Beat, Ekene

Control panel:

shield sword cape helmet

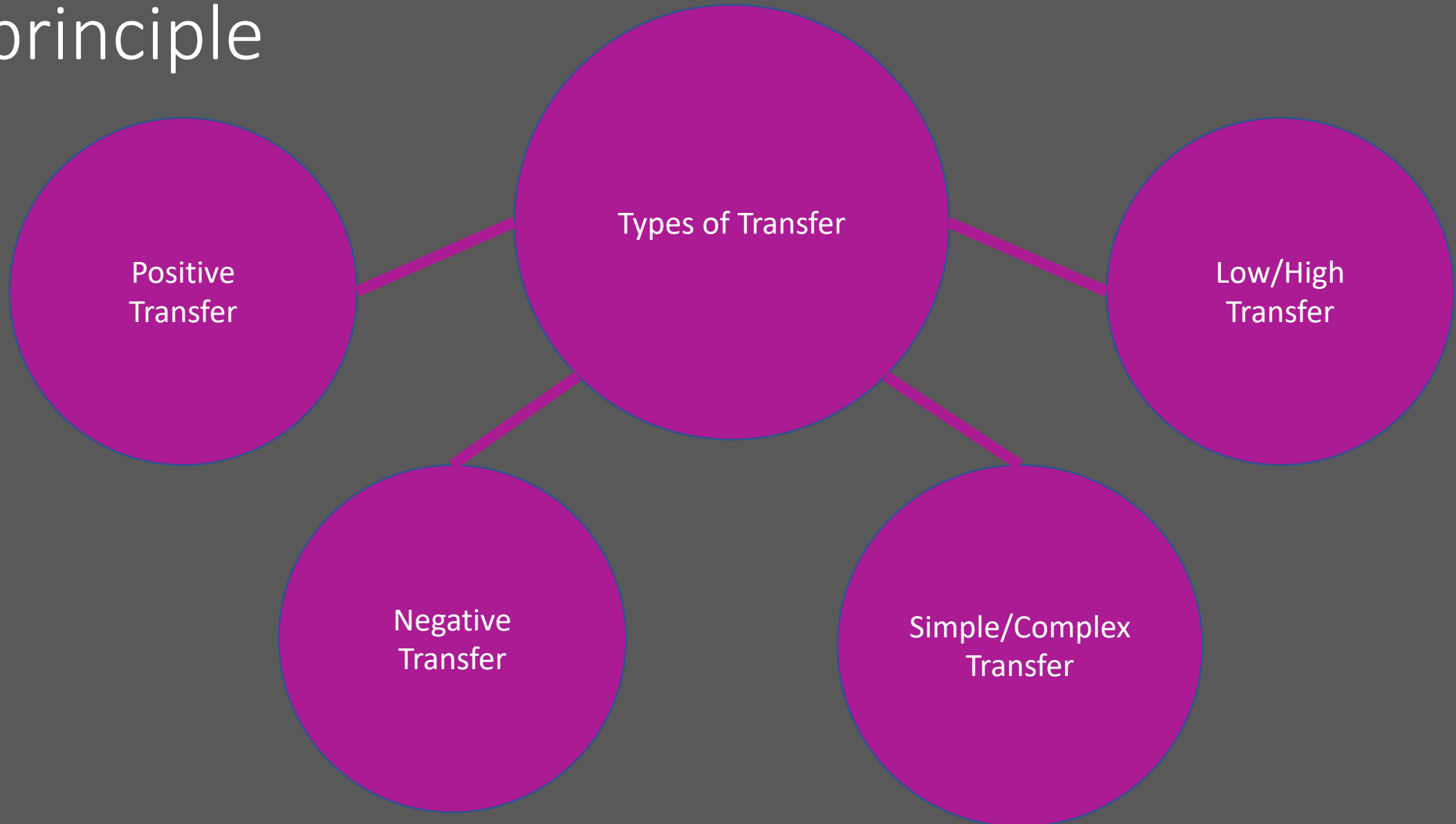
z

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

Specific

Clear and specific outcomes?

Measurable

Define assessment/evaluation to measure outcomes?

Achievable

Is the expected level realistic?

Relevant

Is the goal relevant for the general goals of the learners?

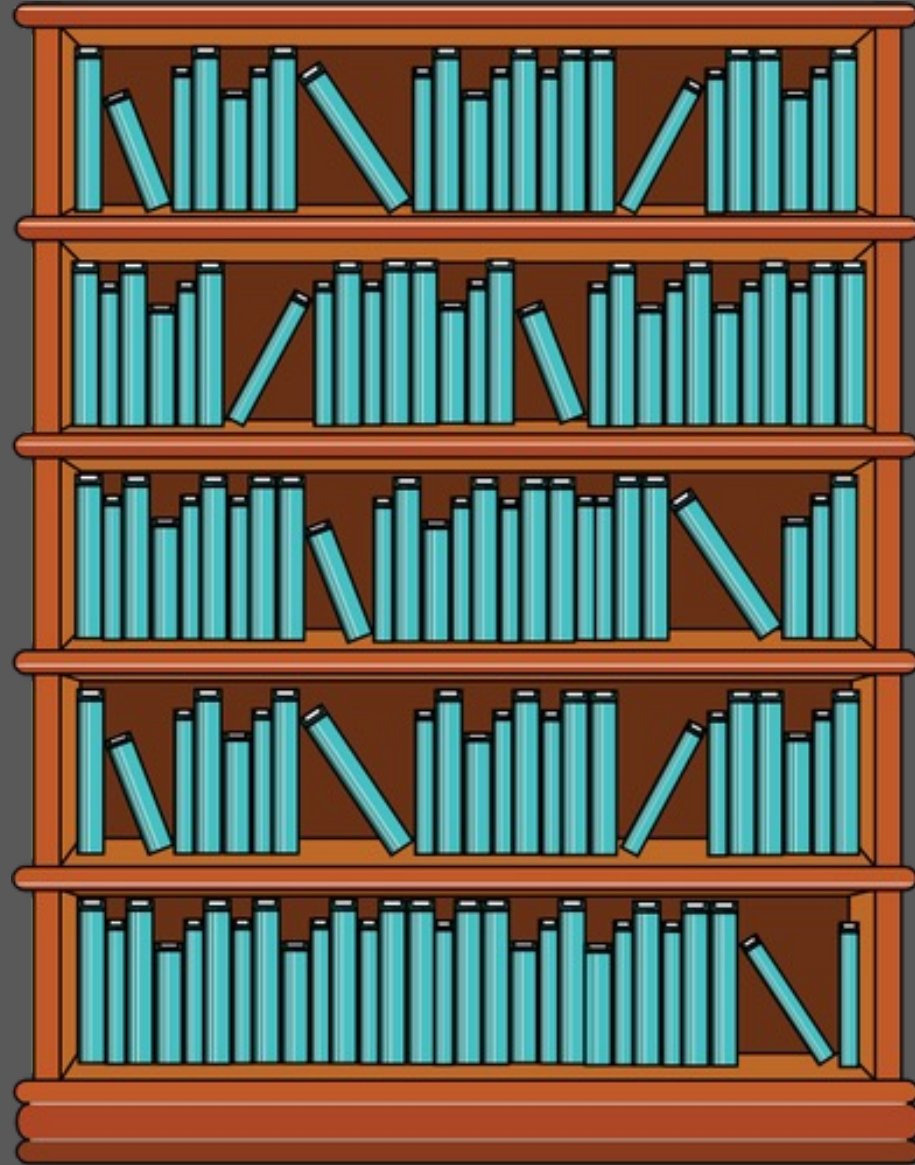
Time-limited

Can it be achieved in a logical time unit?

Keep in mind the expertise reversal principle



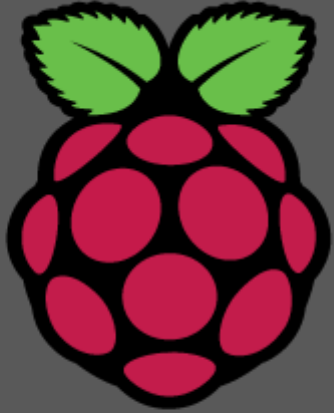
Automated content generation will be a lifesaver



Make your things mobile compatible



Join us and build things



Raspberry Pi



CoderDojo

**HACK
YOUR
FUTURE**

{ Migra Code }
.....