What we’ll be doing

1. Introduction
2. Lego
3. Concepts
4. Examples
Introduction

What do we discuss?

- Problem solving
- Systematic approach
- Mostly imperative programming.

What do we not discuss?

- Language specifics
- Syntax
- C programming
How to think programmatically?

- Systematic Problem Solving
- Algorithms
- Black boxes
Lego!

- All big structures start with small ones
Lego!

- All big structures start with small ones
- Standing on the shoulders of those people who made the Lego bricks
Lego!

• All big structures start with small ones
• Standing on the shoulders of those people who made the Lego bricks
• ”Go build a house”
Lego!

- All big structures start with small ones
- Standing on the shoulders of those people who made the Lego bricks
- "Go build a house"
  - But... Which kinds of bricks are there?
  - But... Which bricks are important?
  - But... Where can I find bricks to suit my needs?
  - But... How do I know if the house is complete? How do I know if it's nice?
Lego!

- All big structures start with small ones
- Standing on the shoulders of those people who made the Lego bricks
- "Go build a house"
  - But... Which kinds of bricks are there?
  - But... Which bricks are important?
  - But... Where can I find bricks to suit my needs?
  - But... How do I know if the house is complete? How do I know if it's nice?
- HOW EVEN DOES BUILDING WORK
Hagrid’s hut: A true story
Black boxes

Something goes in, something else (usually) comes out

output = min(input1, input2)
Black boxes

They work! I don’t usually care how. Unless I have to implement them myself...
## Control, data, concepts

<table>
<thead>
<tr>
<th>Control flow</th>
<th>Data flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipes</td>
<td>Ingredients</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Variables</td>
</tr>
<tr>
<td>Statements</td>
<td>Datastructures</td>
</tr>
<tr>
<td>Function calls</td>
<td>Lists</td>
</tr>
<tr>
<td>if-else</td>
<td></td>
</tr>
<tr>
<td>loops</td>
<td></td>
</tr>
<tr>
<td>Searching</td>
<td>Database</td>
</tr>
<tr>
<td>Arithmetics</td>
<td>Numbers</td>
</tr>
</tbody>
</table>
Recipe/Algorithm

A list of steps to be performed.
Recipe/Algorithm

A list of steps to be performed.

**Recipe**
- Get bowl
- add milk
- add eggs
- add flour
- mix
- bake pancakes

**ATM machine**
- Find account
- check balance
- if sufficient, resolve payment
- else abort payment
If/else

1. if container is empty:
   2. return it to original location
3. else:
   4. leave it out

Jan van Houten, Dinand ten Hooven, Emily Beuken

Programming for Dummies
If container is empty:
    return it to original location
else:
    leave it out

Jan van Houten, Dinand ten Hooven, Emily Beuken
Programming for Dummies
For-loops
For-loops

1. aim at triple20
2. throw
3. aim at triple20
4. throw
5. aim at triple20
6. throw
For-loops

1
2
3
4
5
6

aim at triple20
throw
aim at triple20
throw
aim at triple20
throw

1
2
3

for all 3 darts:
aim at triple20
throw
While-loops
While-loops

1. while there are dishes to be cleaned:
2. pick one of them
3. clean it
Recursion

- Call a function from within that same function
- The object/number/datastructure gets smaller until we can do something with it
- E.g. ripping a piece of paper

```python
def rip(paper):
    if paper is small enough:
        return
    else:
        piece1, piece2 = ripInTwo(paper)
        rip(piece1)
        rip(piece2)
```
Recursion

- Call a function from within that same function
- The object/number/datastructure gets smaller until we can do something with it
- E.g. ripping a piece of paper

```python
def rip(paper):
    if paper is small enough:
        return
    else:
        piece1, piece2 = ripInTwo(paper)
        rip(piece1)
        rip(piece2)
```
Variables/values
Variables/values
Variables/values
Assignment
Assignment

1  

\texttt{tv.channel = 1}
Exercise

To get from waking up in the morning to being on time at university.
Example: Make me a sandwich
Example: Make me a sandwich

```plaintext
makeSandwich

Sandwich makeSandwich(list of ingredients) {
  var plate;
  plate = getBread();
  addIngredients(ingredients);
  giveBreadToMe();
}
```
Example: Make me a sandwich

```java
void addIngredients(List of ingredients) {
    for (ingredient in List) {
        if (find(ingredient) {
            putOnBread(ingredient);
        }
    }
    return;
}
```
Example: Make me a sandwich

```c
void addCheese(Number of Slices) {
    while (bread not full) {
        sliceCheese();
        addSliceToBread();
    }
}
```
Example: Make me a sandwich

```java
void recursiveTomato(Tomato) {
    if (Tomato is slimmer than 2mm) {
        addToBread();
    } else {
        slice1, slice2 <- sliceInHalf(tomato);
        recursiveTomato(slice1);
        recursiveTomato(slice2);
    }
}
```
Example: Going to the uni

- How do you go to the uni? Bicycle, bus...?
- Let’s make an algorithm!
And finally...

- Slides will be online on studcee.svcover.nl
- Please fill in our evaluation forms :)

Jan van Houten, Dinand ten Hooven, Emily Beuken

Programming for Dummies