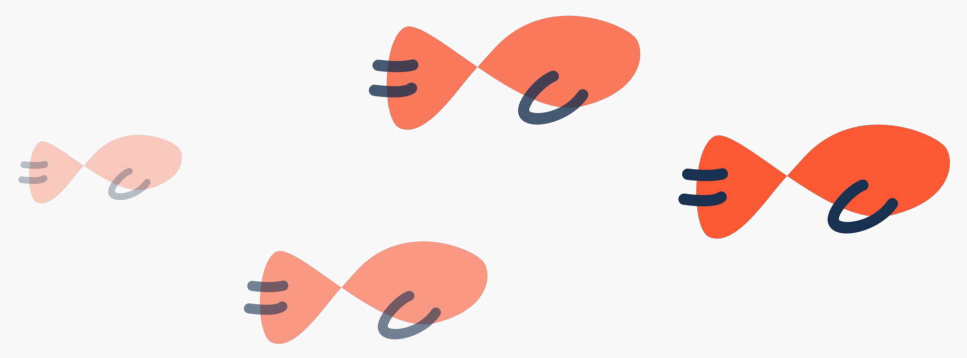
neue fische

School and Pool for Digital Talent

How to design more complex code?



Objective: How to think like a computer without having to talk to the computer

Why:

The acts of figuring out how to structure your program and how to write the actual code are different mental tasks

- Building a program concept is a creative task
- Writing code in the right syntax is detail focused task

It is not effective to do these simultaneously

Thinking like the Computer vs talking to the computer

Programming vs Coding

● It’s about thinking and problem solving	● Pertains to computer code
● Create a set of instructions which defines the application	● Write the instructions for the computer with code
● You need to be creative	● Understanding your program allows you to represent it in code
● The most fundamental skill to have	● Needs to be learned and becomes easier when you have mastered programming
● Isn't technical, but logical. Even kids can program	● Is a technical skill
● An inborn skill, which everyone already has	● You can copy code from Stack Overflow, GitHub, …

Source: https://www.freecodecamp.org/news/programming-coding-developement-whats-the-difference/

Objective: How to think like a computer without having to talk to the computer

What?

There are some ways to map
what a program is meant to do:

- Flowcharts (nodes, eg: https://lucid.app/ or https://app.diagrams.net/)
- Struktograms (aka Nassi-Shneiderman-Diagram, geometric)
- Pseudocode (without specific notations)

Objective: How to think like a computer without having to talk to the computer

What: Pseudocode!

- Informal high-level description
  of a computer program or algorithm.
- An informal guide and a tool
  for thinking through programmatic problems
- Communicates ideas to other people.

Objective: How to think like a computer without having to talk to the computer

How:

- Use simple notation of common programming techniques
- Spell out the instructions step by step
- Balance use of program code and english words, plain text
- Can then be coded in the language of your choice
- When done: Use it as the first comments of your code
  to guide coding and explain what the code is doing

Pseudocode
Common pseudocode notation

- **INPUT** – indicates a user will be inputting something
- **PRINT** – indicates that an output will appear on the screen
- **STORE** – set a variable to a value
- **WHILE** – a **loop** (**iteration** that has a **condition** at the beginning)
- **FOR** – a counting loop (iteration)
- **IF – ELSE** – a decision in which a choice is made
    indent instructions inside a selection or loop

There is no strict set of standard **notations** for pseudocode, these are just common

Example

***First definition** of a program*
that will allow the user to check whether a number is even or odd.

```
IF condition "Number is even" = TRUE
    PRINT "Number is even"
ELSE
    PRINT "Number is odd"
```

Example

Second Iteration *of a program*
that will allow the user to check whether a number is even or odd.

```
IF number % 2 = 0
    PRINT "Number is even"
ELSE
    PRINT "Number is odd"
```

Example

***Third Iteration** of a program*
that will allow the user to check whether a number is even or odd.

```
DEFINE FUNCTION check_even_odd(number):
    IF number % 2 = 0
        PRINT "Number is even"
    ELSE
        PRINT "Number is odd"
```

Example

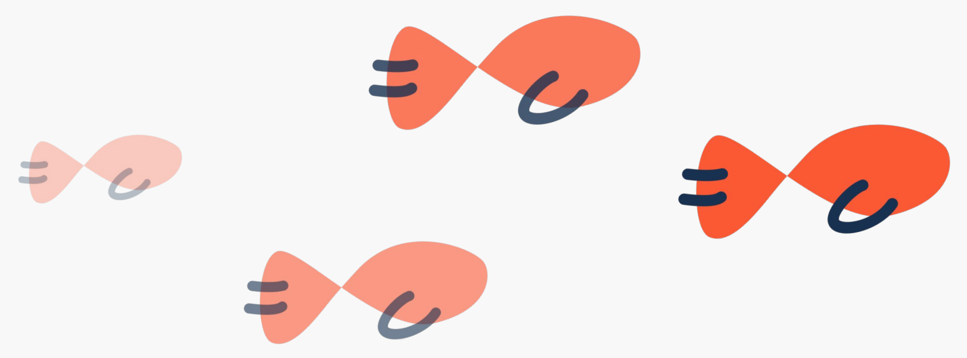
Final Iteration *of a program*
that will allow the user to check whether a number is even or odd.

```
DEFINE FUNCTION check_even_odd(number):
    IF number % 2 = 0
        PRINT "Number is even"
    ELSE
        PRINT "Number is odd"

DEFINE FUNCTION get_input():
    PRINT “What number do you want to check?”
    INPUT user gives a value
    RETURN input value

CALL check_even_odd(get_input())
```

Takeaway!



Make your plan, execute your plan

“There are no technical rules for Pseudocode.
It is meant to be human readable and still convey meaning and flow.”

**Your completed Pseudocode is the plan
that you will use to write your actual code**

Source: Ngunyi Macharia
https://medium.com/@ngunyimacharia/how-to-write-pseudocode-a-beginners-guide-29956242698

Practice

Write a pseudocode for functions that:

1. return the area of a rectangle with sides of which are two given numbers.

2. return the largest of two user entered numbers.

3. return the largest two of three user entered numbers.

4. return the area of circle with radius from output of question 2

5. return the area of a rectangle, the sides of which are the 2 largest of 3 user entered numbers (reuse answer to question 1 and 3)

6. return the sum of the areas of the circles with radius equal to the output of question 3

Let’s think first

Help the Robot to wash the dishes