

Basecamp Dossier – Ruben Flinterman

Name: Ruben Flinterman

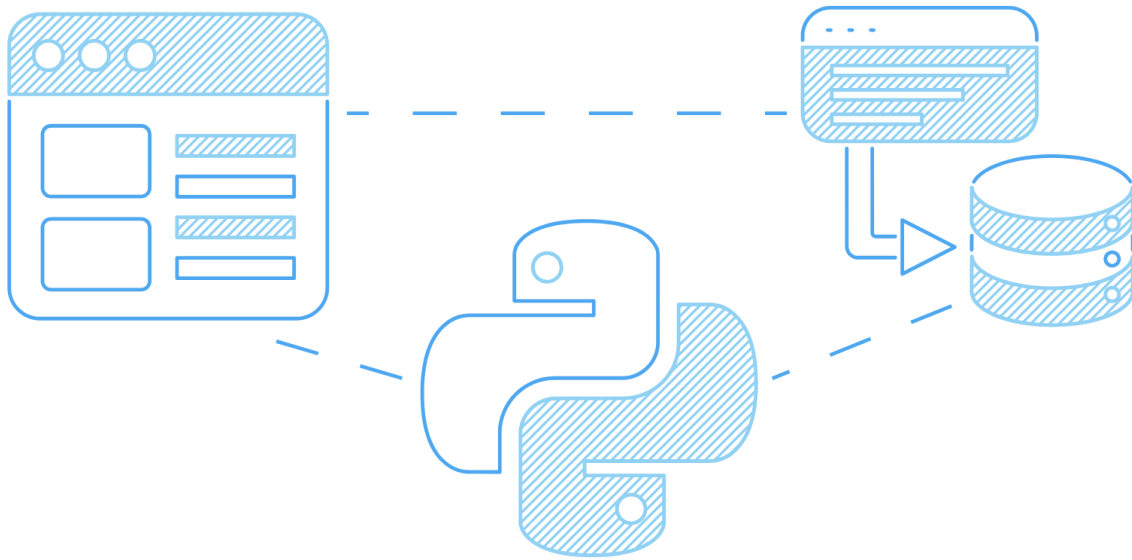
Student number: 1037046

Subject: Basecamp Dossier – Ruben Flinterman

Study Career Coach: Thomas van Iersel

Programming Teacher: Cigdem Okuyucu

Date: 21/12/2023



Preface

Dear reader,

Welcome to my dossier which is essentially my learning journal, documenting the lessons I've picked up during but also next to Basecamp. I've laid out the problems and assignments I tackled, complete with their respective links for quick access to their latest versions. This document also includes my personal goals and feedback I've both received and given.

There are problems which aren't entirely documented as I have previous experience with programming (over 5 years) and I'm meant to write down my growth.

My personal Basecamp experience is that I enjoyed it but everything was relatively easy to figure out and learn if I didn't know it already. The only thing that was harder to do was the third which was also the last challenge week where teams of two, in my case Jonathan Bout and I, needed to make a racing game. This game needed to be made with PyGame which I never touched before which used some interesting functionalities to work. Luckily there were online resources which helped us a bunch on how to tackle some specific issues such as the camera.

Kind regards,
Ruben Flinterman
21/12/2023

Content

Feedback loop

- 1.1 Self-evaluation - week 3
- 1.2 Feedback session: verslag - week 4/5
- 1.3 Plan of approach: learning goals - week 6
- 1.4 Peer-evaluation - week 7
- 1.5 Self-evaluation - week 8
- 1.6 Feedback session: verslag - week 9/10
- 1.7 Plan of approach: learning goals - week 11
- 1.8 Peer-evaluation - week 12
- 1.9 Self-evaluation: end conclusion – week 13/14

Assignments

Arch 1	23
Assignment: A1W1A1 - Year to month & day	23
Assignment: A1W1A2 - Tax & Tip.....	24
Assignment: A1W1P1 - Hello name	25
Assignment: A1W1P2 - Year to month and day	26
Assignment: A1W1P3 - Room area.....	27
Assignment: A1W1P4 - Weight calculation	28
Assignment: A1W1P5 - Four digit sum	29
Assignment: A1W1P6 - Hours, minutes and seconds	31
Assignment: A1W2A1 - Immediate successor	32
Assignment: A1W2P1 - Even or Odd	35
Assignment: A1W2P2 - Leap year.....	36
Assignment: A1W2P3 - Sides to shape	37
Assignment: A1W2P4 - Triangle type.....	38
Assignment: A1W2P5 - Dutch holidays	39
Assignment: A1W2P6 - Dog years	40
Assignment: A1W2P7 - Chessboard colors	41
Assignment: A1W2P8 - Licence plate	43
Assignment: A1W3A1 - Predefined templated.....	46
Assignment: A1W3P1 - Simple palindrome.....	47
Assignment: A1W3P2 - Advanced palindrome	49
Assignment: A1W3P3 - Modular rectangles.....	50
Assignment: A1W3P4 - Celcius to Fahrenheit.....	51
Assignment: A1W3P5 - Multiplication table.....	52
Assignment: A1W3P6 - Binary to Decimal	54

Assignment: A1W3P7 - Truth Tables	55
Arch 2	56
Assignment: A2W5A1 - Processing student data.....	56
Assignment: A2W7A1 - Name hasher.....	59
Arch 3	60
Assignment: A3W09A1 - Car parking.....	60
Assignment: A3W10A1 - Car parking logger.....	61
Assignment: A3W11A1 - Car parking extended	62
Arch 4	63
Assignment: A4W13A1 - Car parking final	63
Assignment: A4W15A1 - Folder structure	66
One of my jobs is as a student researcher for Chemical Technologies at RDM kade (Hogeschool Rotterdam).	67
Learning Kubernetes.....	69
Class based rewrite of Challenge Week 1 in my spare time	71

1. Feedback Loop

1.1 Self-evaluation – Week 3

1. *How do you feel about your choice of course programme after the first few weeks?*

I like the course programme and the friends I made during the first few weeks.

2. *How do you experience Basecamp compared to your expectations beforehand?*

Basecamp is still pretty easy but thanks to that I'm able to rest more. In my case that is really helpful given the fact I've epilepsy.

3. *How do you feel about the class?*

I find the others in the class nice. If you want to have a fun conversation with someone you can.

4. *How do you feel about your learning team?*

The same as with question 3. I like the ones with who I form a team with.

5. *What went well the past few weeks?*

Stuff like learning to know each other, programming and following the lessons went well.

6. *What did not go well the past few weeks? Or was difficult?*

I don't really know what the put here. Everything went pretty well.

7. *What is your step-by-step plan if you get stuck? With coding or studying.*

- 1) What went wrong
- 2) Why did it go wrong
- 3) Is there a solution (research)
- 4) Can I solve it myself or do I need help?
- 5) Is the problem solved, nice, if not back to step 1.

8. *What resources do you have available for Basecamp?*

Depends on what you call resources, but I would say:

- 1) You've internet
- 2) You've a learning team
- 2) You've teachers and peer-coaches who can help you if need be

9. *What do you think about the content of Basecamp?*

Personally I think that it's nice to have documentation on Github but almost no one knows what Github is. The steps listed on Github are long as well which makes me 'skim' the steps instead. (At the end of the road I'm sure that I know how and what to do as I learned a lot in the past and do look for and read keywords.)

10. *How do you feel about learning to program in Python?*

Going to be honest, I personally kind of dislike Python even though some assignments are fun to do. I do like Python more now but languages like C++, C# or even Java and Javascript are in my opinion more fun (and useful) as they use and share different expressions (in relation to Python).

You have completed the study habits and skills checklist.

1. *What was the result?*

If I remember correctly, it was a 179

2. *To what extent do you recognize yourself in the results?*

Besides note taking which I don't often do (or at least, not physically) I do recognize myself in the result.

1.2 Feedback session: verslag - week 4/5

3 What things are going well according to your teacher?

I'm finishing the assignments on time while learning stuff I need to know.

4 Did you discuss things that could/should be improved? If so, what are they?

My dossier might have to little text around the topic 'what did I learn?', but I should be able to fix that if I also mention the projects I'm doing next to school and how I did/am doing those.

5 Did you and your teacher make an agreement/agreements for the time coming?

I'm allowed to skip problems and make the assessments so I don't have to wait if there are no assessments to make anymore. If it turns out there are no assessments available and/or want to spend more time on coding I can do 'Advent of Code'.

6 How do you feel about the feedback received? Do you recognise the points that were discussed?

Yes, I recognize myself in the feedback.

1.3 Plan of approach: learning goals - week 6

My learning goal professional skills:

'I will/am improving my writing skills during basecamp. I already began in Arch 1 but at the end of Arch 4.'

Specific	What is the problem or situation? What do you want to develop when it comes to professional skills? I want to learn to improve my writing and presentation skills in the sense that people who barely or don't know anything from computer science can easily understand what I'm saying.
Measurable	When will you know you have realised your goal? How can others see this? E.g. How often am I going to do it? How many hours will I spend on it? When I get the feedback from people who read my text or follow a presentation of mine that they could easily follow me.
Acceptable	Do you stand behind your learning goal and do you have sufficient motivation for it? Is the goal achievable? What qualities and skills can you use to achieve the learning goal?? Both writing and presenting are not the most fun part in my opinion so I don't have a lot of motivation but the goal is achievable, especially since I can talk with others pretty easily. The presenting it self wouldn't be the problem, but the way I format information is.
Realistic	Is my learning goal realistic? Do I have the knowledge, resources to achieve my learning goal? How can others possibly help you achieve your learning goal? Who is/are they? Is the learning goal not too simple? I do think the learning goal is realistic. Others could help me achieve my goal by giving constructive feedback. In my case others would be the people who would receive the information I am trying to convey. I don't think the learning goal is not too simple as it is a habit of mine to use words which are used a lot in our work field. – Programmers can easily follow me, but I'm afraid clients would not entirely follow me.
Time	Within what time do you want to achieve your learning goal? At what times will I work on my learning goal? The writing part is something I want to improve in a relatively short period of time. Given that we need to write our dossier that must be possible. I'm not sure about the presentation(s). This might be a good thing to improve during an internship.

My learning goal (learning) programming:

'To rewrite my code less unless it is absolutely necessary. This way I should be able to train myself to think less difficult.'

Specific	What is the problem or situation? What do you want to develop when it comes to professional skills? I am used to think too difficult regarding solutions. I want to 'reset' my thinking pattern in order to solve some solutions without rewriting code for optimization.
Measurable	When will you know you have realised your goal? How can others see this? E.g. How often am I going to do it? How many hours will I spend on it? I know when I realized my goal when I write less code without rewriting it. Although this is a long-term goal it will be a goal which I could spend at least 10 hours on per week. During basecamp I would probably spend more time it so it would most likely be a minimum of 15 hours.
Acceptable	Do you stand behind your learning goal and do you have sufficient motivation for it? Is the goal achievable? What qualities and skills can you use to achieve the learning goal?? I've more than sufficient motivation and I do think the goal is achievable. The qualities and skills I could/would use would be my programming skills so I can use assessments and hobby+work projects to achieve my set goal.
Realistic	Is my learning goal realistic? Do I have the knowledge, resources to achieve my learning goal? How can others possibly help you achieve your learning goal? Who is/are they? Is the learning goal not too simple? Yes, I do think my goal is achievable and I do have the knowledge and resources to achieve my goal. Others could potentially give feedback and ask why I solved something a certain way. The difficulty of this goal is moderate.
Time	Within what time do you want to achieve your learning goal? At what times will I work on my learning goal? It will be a goal which I could spend at least 10 hours on per week. During basecamp I would probably spend more time on it so it is most likely a minimum of 15 hours.

1.4 Peer-evaluation - week 7 (Part 1)

Part A Peer evaluation

1. Give some of the do's and don'ts of giving and receiving feedback.

Do's:

1. Give both positive and negative feedback
2. Make sure the feedback is constructive
3. Listen to the feedback you receive
4. Process the feedback you receive

Don'ts:

1. Don't only give negative feedback
2. Don't give non-constructive feedback. The other can't do anything with that.
3. Don't ignore the feedback. The feedback wasn't given for nothing.

2. Describe for each team member at least one thing he/she does well and at least one point for development.

This is what you do well:

Julian: He works seriously and picks up the material quickly. He also is open to help anyone who is stuck at coding.

Bente: She is really motivated, works hard as well, and stays if she thinks she doesn't get the material yet. She asks for help when she needs it.

Charlize: The same as with Bente. She is motivated, works hard as well, and she looks for help when she needs it.

Arthur: I think that Arthur is very determined and works hard.

This is what I see as a point for development:

Julian: Your power might be your weakness. You're really good at working hard (by being motivated) but be careful during (big) projects and take some 5 min breaks.

Bente: Maybe read the documentation a bit more and combine that with your questions.

Charlize: You don't always sit with the learning group so I can't give any criticism right now.

Arthur: Make sure to attend the Teams meetings.

1.4 Peer-evaluation - week 7 (Part 2)

Part B Code review

1. **Find a student from a different learning team and decide together which programming assignment from week 3, 5 or 6 you are going to review. Look at the code of your fellow student en answer the following questions:**
 2. **Which programming assignment did of which student did you review?**
Programming assignment: A2W6A1 – Addressbook
Student: Hidde Sterken
 3. **Do the variables have correct names? (is it a description of what it contains? Which one(s) would you name differently? Are there obsolete variables? How does this code compare to your code?)**
The variables were rather short but the code was optimized.
 4. **Does the code do what it's supposed to do? And is the code clear in what it does? Doet de code wat het moet doen? En is de code duidelijk in wat het doet? How does this compare to how you solved it?**
The code does what it is supposed to do and it's clear from how the methods are written what does what.
 5. **How do the if/while statements look? (think about: are they clear and clean, are there obsolete statements?) How does this compare to how you solved it?**
They look good, although he used try and except more then I would use it.
 6. **Are the PEP8 guidelines for Python applied? Where does this go well? Where can it be improved?**
Not all guidelines are used. For example, variable naming could be improved
 7. **Is there a simpeler solution possible? Compare your code with that of your fellow student. What differences and similarities do you see?**
Probably, yes, but seeing from what we have learned in class it's really good.
 8. **How is a wrong input handled? (if applicable to the assignment you chose**
With if/else/elif statements.
 9. **Write a short summary of your findings in Part B question 3 to 8. What goes well? What can be improved? What differences and similarities do you notice when you compare your code with the code of your fellow student?**
Although both scripts work I believe my code is more dynamic if you compare my code with Hidde's. Both scripts use methods and are optimized, all be it in a different way.

1.5 Self-evaluation - week 8

1. How do you feel about your choice of study program now?

I feel like it was a good choice but I do find a lot of stuff easy and find myself bored sometimes.

2. How do you experience Basecamp compared to your expectations?

I expected all assignments were mandatory but only the assessments are. This does makes it easier then expected (as the only constraint is sometimes time).

3. What went well in recent weeks? Or what are you proud of?

I was able to do the assessments without the problems, as of now, never got a mark under the 8 and have enough time to rest during the week(end).

4. What hasn't gone so well in recent weeks? Or what was difficult?

As mentioned earlier, time was sometimes a constraint, but I managed to do everything well.

5. To what extent does the schedule of Basecamp match your study pace?

It matches well. Before Informatica/computer-science I never really had free time as I filled in almost every gap. Now I take more free time in the weekend and during the week which helps me rest. Needed especially because of my epilepsy.

Evaluation learning objectives

6. What learning objectives have you set?

a. Professional Skills:

Get better at both presenting and writing documentation as I want to get better at communicating with people who can't program.

b. Computer programming:

Because I have programming experience I tend to make my solutions to complicated. I want to train myself to make it less complicated when it doesn't need to be complicated.

7. What activities did you undertake to work on the learning objectives?

- 1) Read alternate solutions on the internet and looked at the code of others.
- 2) Rewrote my code in a couple of ways (replaced old code) to both optimize it on length and simplicity

8. What could you possibly do in addition to what you have done so far? Or what could you do differently?

As of now, I don't know.

1.6 Feedback session: verslag - week 9/10

1. ***What things are going well according to your teacher?***

I'm making good progress with my assignments and problems and ask for feedback when I think I need it.

2. ***Did you discuss things that could/should be improved? If so, what are they?***

I don't always use a correct writing format. For example, I talk about person X but I forgot to properly introduce them. I personally think it's because I tend to write at a relatively quick speed for a concept and I want to make sure the concept is good before finalizing the text.

3. ***Did you and your teacher make an agreement/agreements for the time coming?***

No, we did not.

4. ***How do you feel about the feedback received? Do you recognise the points that were discussed?***

I recognize myself in the feedback, as I already had the personal goal of improving writing before I received the feedback.

1.7 Plan of approach: learning goals - week 11

My learning goal professional skills:

'To get/am going to be better at scheduling so I can be more productive.'

My programming learning goal:

'I'm going to think less difficult and rewrite less code as most of the time it isn't necessary'

Specific	What is the problem or situation? What do you want to develop when it comes to professional or programming skills? Even though I'm relatively good at it I want to get better at writing documentation as I find it an important part of programming.
Measurable	When will you know you have realised your goal? How can others see this? E.g. How often am I going to do it? How many hours will I spend on it? I'm going to spend time on it during my work so probably around 4-6 hours a week and I will know that I reached my goal if I can write good documentation at a relative fast pace and others can understand what I'm trying to convey. Others will know I reached my goal if they can understand my documentation and find it relatively easy to resume the work I did.
Acceptable	Do you stand behind your learning goal and do you have sufficient motivation for it? Is the goal achievable? What qualities and skills can you use to achieve the learning goal?? I stand behind my goal and I do think the goal is reachable. The motivation is there but I'm not fully sure if I can spend the amount of hours on it that I want to spend on it because of my busy life.
Realistic	Is my learning goal realistic? Do I have the knowledge, resources to achieve my learning goal? How can others possibly help you achieve your learning goal? Who is/are they? Is the learning goal not too simple? As partially said in 'Acceptable' I do think it's reachable and therefore realistic. I do have the knowledge and 'resources' such as people whom I can ask for feedback. The goal is not too simple as I tend to want things to go relatively smoothly, so that includes writing documentation.
Time	Within what time do you want to achieve your learning goal? At what times will I work on my learning goal? I want to be good at it at the end of the school year (1 st of September). The reason why I choose for this is because I then have enough time to spend on it as I won't be able to always write documentation.

1.8 Peer-evaluation - week 12

Part A Peer evaluation

1. ***How or with what knowledge/skill can you help/support other team members?***

I can help others with how you can apply object oriented programming.

2. ***With what could you use help/support from other team members?***

To be honest, I'm not so sure.

3. ***Describe for each team member at least one thing he/she does well and at least one point for development.***

This is what you do well:

Jonathan Bout: I noticed that Jonathan is really good at problem solving and also really determined to solve the issue at hand.

This is what I see as a point for development:

Jonathan Bout: During the challenge weeks I did with Jonathan I noticed he can sometimes be a bit quiet both physically and even more so digitally which can cause confusion on what he works on. When we introduced a Kanban board this seemed to be more of a solution.

4. ***Discuss the feedback you wrote down in question 2 with your team members. Write down what feedback you received from your team members.***

This is the feedback I received from my learning team (write down multiple items):

This is what I do well:

These are points for development:

Part B Code review

1. **Find a student from a different learning team and decide together which programming assignment from week 9, 10 or 11 you are going to review. Look at the code of your fellow student and answer the following questions:**

2. **Which programming assignment did of which student did you review?**

Programming assignment: 9

Student: Jonathan Bout

3. **Do the variables have correct names? (is it a description of what it contains? Which one(s) would you name differently? Are there obsolete variables? How does this code compare to your code?)**

All variables are correct have have a datatype assigned. I don't see any obsolete variables. The code compared to mine is that my code don't all have a datatype assigned.

4. **Does the code do what it's supposed to do? And is the code clear in what it does? Doet de code wat het moet doen? En is de code duidelijk in wat het doet? How does this compare to how you solved it?**

The code works as how it is supposed to work. The print lines are also perfectly organized in a way that it is both readable while looking and running the code.

5. **How do the if/while statements look? (think about: are they clear and clean, are there obsolete statements?) How does this compare to how you solved it?**

They are pretty clean and I don't really see any unnecessary if and/or while statements.

6. **Are the PEP8 guidelines for Python applied? Where does this go well? Where can it be improved?**

Yes the code uses PEP8 guidelines but I would have liked if Jonathan had added docstring and comments.

7. **Is there a simpeler solution possible? Compare your code with that of your fellow student. What differences and similarities do you see?**

The code is pretty simple already. Some similarities I'm seeing is the way we handle the (beginning of) check_in. The difference is that I use isinstance to make sure if variables have the correct type stored and he doesn't do that.

8. **How is a wrong input handled? (if applicable to the assignment you chose)**

He uses an if and else statement to check if the correct input is given.

9. **Write a short summary of your findings in Part B question 3 to 8. What goes well? What can be improved? What differences and similarities do you notice when you compare your code with the code of your fellow student?**

The code is clean and does the job. It's comparable with my own code but Jonathan clearly spends time on how to make the code shorter which I didn't spend a lot of time on. He uses a match case instead of an elif for the menu which works well although this wouldn't be my first choice.

1.9 Self-evaluation: end conclusion – week 13/14

A. Evaluation learning goals

Professional Skills

Name here both of your learning goals for professional skills:

- **Learning objective week 6:**
"Improve my writing and presentation skills"
- **Learning objective week 11:**
"Getting better at writing documentation" and "To get/am going to be better at scheduling so I can be more productive."

We would like to get a good picture of your development regarding *professional skills*. Answer the questions below. Provide good evidence. **Provide at least 1, no more than 2 pieces of evidence that support your development.**

1. ***Looking at your learning goals related to professional skills, how have you developed on these goals? What results have you achieved? What succeeded and what didn't (completely)? What has noticeably/visibly improved and/or changed for yourself and others?***

I'm now able to write faster with a relatively professional language in the timeslots I planned for it. People seem to understand my writing and Kanban boards and my Outlook Calendar are my go-to, to plan things I should do or go to.

2. ***To what extent was your approach effective? What could you have improved (with your current understanding) in your approach?***

My approach was fairly effective as my planning regarding where I should be is helping me remember a lot. But the kanban boards seem to be really good for programming tasks only at this moment in time.

3. ***What will you take with you from what you learned into the future? (e.g., to the 2nd semester?)***

I will take my writing and planning experience with me as that proved useful in the last few months.

Programming

Name here both of your learning goals for professional skills:

- **Learning objective week 6:**
"To rewrite my code less unless it is absolutely necessary. This way I should be able to train myself to think less difficult."
- **Learning objective week 11:**
"I'm going to think less difficult and rewrite less code as most of the time it isn't necessary"

We would like to get a good picture of your development regarding *programming*. Answer the questions below. Provide good evidence. **Provide at least 1, no more than 2 pieces of evidence that support your development.**

1. ***Looking at your learning goals related to programming, how have you developed on these goals? What results have you achieved? What succeeded and what didn't (completely)? What has noticeably/visibly improved and/or changed for yourself and others?***

As this was a tendency which is kind of hard to 'unlearn' I forced myself to write the code a maximum of 2 times and think it out more properly by adding the base functions/methods first, then comments of what it should do and then actually writing the code. This helped a lot and extended my existing knowledge of why planning can be really important.

2. ***To what extent was your approach effective? What could you have improved (with your current understanding) in your approach?***

The approach was effective as I was rewriting code less with the inline comments I placed beforehand. By no means it was preventing it but it was/is a good start in my opinion.

3. ***What will you take with you from what you learned into the future? (e.g., to the 2nd semester?)***

I'll take the newfound approach with me. Added comments beforehand to empty functions/methods was helping me both thinking out and structuring out my code.

B. Evaluation learning goals

Professional Skills

Student success

- ***What am I already good at?***
I'm good at finding the 'drive' to learn new things.
- ***What can I still improve?***
I should try to plan brakes in for myself. Learning is fun but applying new knowledge isn't doable in the long run if you are tired.

Research

- ***What am I already good at?***
Finding the right resources to finish my tasks, or, if the solution doesn't exist finding the correct place to ask my specific question where I can get a good answer.
- ***What can I still improve?***
It might be useful to document my research in a better way.

Problem solving

- ***What am I already good at?***
Thinking outside the box by using my creative side.
- ***What can I still improve?***
Not making it myself difficult. Being creative is fun but some things are not as difficult as they seem.

Programming

Basic and advanced structures

- ***What am I already good at?***
Programming in an organized manner by using objects and inheritance.
- ***What can I still improve?***
I'm not really good at using datatypes such as unsigned ints yet.

Debugging and testing

- ***What am I already good at?***
Writing automated tests and debugging them.
- ***What can I still improve?***
The forgetfulness to write the actual tests in the first place.

Data processing

- ***What am I already good at?***
Writing API endpoints
- ***What can I still improve?***
Regarding data processing? I'm not sure.

C. Evaluation BaseCamp

1. ***Describe here how you look back on Basecamp. How did you experience it? What did you learn along the way?***

I found Basecamp itself really easy even though I barely touched Python in the past. So I was glad there was the occasional Challenge Week so I could change pace for a few days. After a few weeks I stopped doing the problems as they weren't mandatory and I understood how to make the assessments. I did notice however that Codegrade doesn't want to play nice with a lot of the time by giving useless errors or looking too strict at `print()` statements.

2. ***What are you most proud of (both personally and technically)?***

Personal: Making a game using Pygame in less than a week without previous experience with Pygame.

Technical: Figuring out how OpenCV works so another student and I were able to implement dynamic track hitboxes. (Because game objects are a 'rect' in Pygame you can't easily make fancy shapes with only Pygame)

3. ***Give a tip and a top regarding Basecamp.***

Tip: Stop using Codegrade and start utilizing normal IDEs such as PyCharm and learning the student's GIT. This way it's also easier to do projects with other students.

Top: The challenge weeks were fun and give you an extra EC if you finish them which is nice and for your dossier you can also document what you learned outside of school which personally gave me a good extra way to document that I am able to learn other languages without saying "I didn't learn anything" in my documentation.

2 Codegrade

2.1 Assignments & problems

In week 5 I started to skip the problems and fully focus on the assignments and dossier hence why some links are not provided.

The last few assignments I made (week 14 and 15) the grades are lower then normal. That's because I was focusing more on the dossier so I wanted to spend less time on perfecting my code.

Assignment description	CodeGrade-link
A1W1A1 - Year to month & day	https://app.codegra.de/courses/5199/assignments/38788/submissions?search=1037046
A1W1A2 - Tax & Tip	https://app.codegra.de/courses/5199/assignments/38789/submissions?search=1037046
A1W1P1 - Hello name	https://app.codegra.de/courses/5199/assignments/38796/submissions?search=1037046
A1W1P2 - Year to month and day	https://app.codegra.de/courses/5199/assignments/38797/submissions?search=1037046
A1W1P3 - Room area	https://app.codegra.de/courses/5199/assignments/38798/submissions?search=1037046
A1W1P4 - Weight calculation	https://app.codegra.de/courses/5199/assignments/38799/submissions?search=1037046
A1W1P5 - Four digit sum	https://app.codegra.de/courses/5199/assignments/38800/submissions?search=1037046
A1W1P6 - Hours, minutes and seconds	https://app.codegra.de/courses/5199/assignments/38801/submissions?search=1037046
A1W2A1 - Immediate successor	https://app.codegra.de/courses/5199/assignments/38790/submissions?search=1037046
A1W2P1 - Even or Odd	https://app.codegra.de/courses/5199/assignments/38802/submissions?search=1037046
A1W2P2 - Leap year	https://app.codegra.de/courses/5199/assignments/38803/submissions?search=1037046
A1W2P3 - Sides to shape	https://app.codegra.de/courses/5199/assignments/38804/submissions?search=1037046
A1W2P4 - Triangle type	https://app.codegra.de/courses/5199/assignments/38805/submissions?search=1037046
A1W2P5 - Dutch holidays	https://app.codegra.de/courses/5199/assignments/38806/submissions?search=1037046
A1W2P6 - Dog years	https://app.codegra.de/courses/5199/assignments/38807/submissions?search=1037046
A1W2P7 - Chessboard colors	https://app.codegra.de/courses/5199/assignments/38808/submissions?search=1037046
A1W2P8 - License plate	https://app.codegra.de/courses/5199/assignments/38809/submissions?search=1037046
A1W3A1 - Predefined templates	https://app.codegra.de/courses/5199/assignments/38791/submissions?search=1037046
A1W3P1 - Simple palindrome	https://app.codegra.de/courses/5199/assignments/38810/submissions?search=1037046
A1W3P2 - Advanced palindrome	https://app.codegra.de/courses/5199/assignments/38811/submissions?search=1037046

A1W3P3 - Modular rectangles	https://app.codegra.de/courses/5199/assignments/38812/submissions?search=1037046
A1W3P4 - Celsius to Fahrenheit	https://app.codegra.de/courses/5199/assignments/38813/submissions?search=1037046
A1W3P5 - Multiplication table	https://app.codegra.de/courses/5199/assignments/38814/submissions?search=1037046
A1W3P6 - Binary to Decimal	https://app.codegra.de/courses/5199/assignments/38815/submissions?search=1037046
A1W3P7 - Truth tables	https://app.codegra.de/courses/5199/assignments/38817/submissions?search=1037046
A2W5A1 - Processing student data	https://app.codegra.de/courses/5199/assignments/38792/submissions?search=1037046
A2W5P1 - Automated arithmetics	
A2W5P2 - Taxi Fares	
A2W5P3 - Triangle Checker	
A2W5P4 - Integer checker	
A2W5P5 - Simple Password Generator	
A2W5P6 - Twelve Days of Christmas	
A2W6A1 - Addressbook	https://app.codegra.de/courses/5199/assignments/38793/submissions?search=1037046
A2W6P1 - Unique Characters	
A2W6P2 - Book Information	
A2W6P3 - Valid Password Checker	
A2W6P4 - Average Temperatures	
A2W6P5 - Morse Code Translator	
A2W7A1 - Name hasher	https://app.codegra.de/courses/5199/assignments/38794/submissions?search=1037046
A2W7P1 - Daily Temperatures Amsterdam	
A2W7P2 - Netflix titles	
A3W09A1 - Car parking	https://app.codegra.de/courses/5199/assignments/38795/submissions?search=1037046
A3W09P1 - Car dealer program	
A3W09P2 - Product shop	
A3W09P3 - Password manager	
A3W09P4 - Distance Converter	
A3W10A1 - Car parking logger	https://app.codegra.de/courses/5199/assignments/38816/submissions?search=1037046
A3W10O1 - File line numbers	
A3W10O2 - Word to password generator	
A3W10O3 - Repeating word detector	
A3W10O4 - Sensitive word replacer	
A3W10P1 - Python head program	
A3W10P2 - Python tail program	
A3W10P3 - Longest word identifier	
A3W10P4 - Word occurrences	
A3W10P5 - Comments remover	
A3W10P6 - Comments checker	

A3W11A1 - Car parking extended	https://app.codegra.de/courses/5199/assignments/38830/submissions?search=1037046
A3W11P1 - Movie collection	
A3W11P2 - Banned video games	
A4W13A1 - Car parking final	https://app.codegra.de/courses/5199/assignments/38850/submissions?search=1037046
A4W13P1 - Student database	
A4W13P2 - Bookstore	
A4W14A1 - Name hasher 2.0	https://app.codegra.de/courses/5199/assignments/38853/submissions?search=1037046
A4W14P1 - Sorting strings	
A4W14P2 - Code performance	
A4W15A1 - Folder structure	https://app.codegra.de/courses/5199/assignments/38857/submissions?search=1037046
A4W15P1 - Positive numbers recursion	
A4W15P2 - Factorial	
A4W15P3 - Find in list	
A4W16M1 - Final Project	https://app.codegra.de/courses/5199/assignments/38860/submissions?search=1037046

2.2 Class learning activities

2.2.1 Arch 1

	CodeGrade-link
A1W4L1 - Learning Activity	https://app.codegra.de/courses/5199/assignments/42571/submissions?search=1037046
A1W4L2 - Learning Activity	
A1W4L3 - Learning Activity	

2.2.2 Arch 2

Assignment description	CodeGrade-link
A1W8L1 - Learning Activity	https://app.codegra.de/courses/5199/assignments/44541/submissions?search=1037046
A1W8L2 - Learning Activity	
A1W8L3 - Learning Activity	

2.2.3 Arch 3

Assignment description	CodeGrade-link
A1W12L1 - Learning Activity	https://app.codegra.de/courses/5199/assignments/48791/submissions?search=1037046
A1W4122 - Learning Activity	
A1W4123 - Learning Activity	

3 Appendices

3.1 Overview of weekly activities of Basecamp

Arch 1

Assignment: A1W1A1 - Year to month & day

Creation Date: 04-09-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

I wanted to have a variable for each calculation and since the input will be the year I've casted that to an int and called the variable 'y'.

When printing I used 3 loose print() methods to prevent one big line and to make the code a bit more readable. This is how it would have looked when I put everything on one line:

```
print("Years: " + str(y) + "Months: " + str(m) + ", Days: " + str(d))
```

Code Snippet

```
# Print what you need for the input
print("Enter amount of years:")

# Request input and convert it to an int so we can use it in a formula
y = int(input())

# Use the input (y) to calculate months and days
m = y * 12
d = y * 365

# Print the results - variable should be string since we concatenate it to a string
print("Years: " + str(y))
print("Months: " + str(m) + ",")
print("Days: " + str(d))
```

Assignment: A1W1A2 - Tax & Tip

Creation Date: 04-09-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

Because I needed decimals and didn't want to write more code than necessary I cast the input to float and force the decimal that way.

Then when printing I use `%.3f` to say I want to print 3 decimals if the outcome of the calculations provide a 3 decimal outcome.

Code Snippet

```
# Request input and convert it to a float so we can use it in a formula with decimals
price = float(input())

# Calculate tip, tax and total
tip = price / 100 * 15
tax = price / 100 * 21
total = price + tip + tax

# Print the results - Instead of converting the variable to string we say that we expect a 3 decimal float
print("Tax: %.3f" % tax)
print(", Tip: %.3f" % tip)
print(", Total: %.3f" % total)
```


Assignment: A1W1P1 - Hello name

Creation Date: 04-09-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

The input method defaults to strings, so I concatenated the input which I have called name, to "Hello" in the print method to successfully pass the assignment.

Code Snippet

```
# Print what we need
print("What's your name?")

# Request input
name = input()

# Print input - no conversion needed for the variable as it's already a string
print("Hello " + name)
```

Assignment: A1W1P2 - Year to month and day

Creation Date: 04-09-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

I wanted to have a variable for each calculation and since the input will be the year I've casted that to an int and called the variable 'y'.
When printing I used loose print() methods to prevent one big line and to make the code a bit more readable.

Code Snippet

```
# Print what we need
print("Enter amount of years:")

# Request input and convert it to an int so we can use it in a formula
y = int(input())

# Use input to calculate both months and days
m = y * 12
d = y * 365

# Print the result - variable should be string since we concatenate it to a string
print("Months: " + str(m))
print(", Days: " + str(d))
```

Assignment: A1W1P3 - Room area

Creation Date: 04-09-2023

What did I learn?

I learned nothing with this assignment.

Why/how did I solve it?

I wanted to print what I needed separately from the input() method so I basically repeat the first `print()` and `int(input())` lines.

Then, since I want to calculate the area I just do `a = l*w` (meaning `area = length * width`) and print the result (`a`)

Code Snippet

```
# Print what you need
print("Enter Width:")

# Request input and convert it to an int so we can use it in a formula
w = int(input())

# Print what you need
print("Enter Length:")

# Request input and convert it to an int so we can use it in a formula
l = int(input())

# Calculate area with the width and height input we just gathered
a = l*w

# Print the result - variable should be string since we concatenate it to a string
print("The Area of the Room: " + str(a))
```

Assignment: A1W1P4 - Weight calculation

Creation Date: 04-09-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

I've created two variables (being `widget_height` and `gizmo_height`) as these values will never change and will be used globally (hence why they are at the top).

I then followed up, two times, with the method `print()` and the method `input()` to determine how many widgets and gizmos we have.

When that's determined we calculate the weight total for the widgets and gizmos (`widget_total_weight` and `gizmo_total_weight`).

Then to fulfill the requirements of the assignment we add both **individual totals** to get the final total. Now if we print `total` we have fulfilled the requirements.

Code Snippet

```
# Widget and Gizmo height are static
widget_weight = 75
gizmo_weight = 112

# Request input and convert it to an int so we can use it in a formula
widget_amount = int(input())

# Request input and convert it to an int so we can use it in a formula
gizmo_amount = int(input())

# Calculate the total weight from both Widget and Gizmo
widget_total_weight = widget_weight * widget_amount
gizmo_total_weight = gizmo_weight * gizmo_amount

# Now add both weights so we have the total weight
total = widget_total_weight + gizmo_total_weight

# Print the total weight - variable should be string since we concatenate it to a string
print("The Total Weight of the Order:" + str(total))
```

Assignment: A1W1P5 - Four digit sum

Creation Date: 04-09-2023

What did I learn?

I learned that you can convert a string to an array using

```
input_to_array = [char for char in input_to_str]
```

How did I learn it?

Not applicable

Why/how did I solve it?

To convert a string full of numbers (Ex: 1234) to a sum ($1+2+3+4=10$) I decided that it might be the best way to program

the software in 'steps'. Meaning that each variable, loop or statement will be limited in function but also always finish one task - an exception being the first variables which are needed to store data and therefore won't be doing any 'tasks' (i.e calculations).

The best way I could think of solving this is to treat the string as an array (in a loop) and then use `[]` to index that position.

I consider this a good approach because this way, again, you only have to do one 'task' to accomplish adding the total of the sum as well as actually 'writing' the formula.

To 'write' the sum we check in each iteration if the storage we are going to write to is odd. If it is odd we'll add a `+` sign first and then the number. If it is anything else then odd (that would be even) we only add the number.

This should be done because we never want to begin with a `+` sign but always want it after the first number for example.

Then, to actually calculate the final result we do a slight alteration/simplification of the code we made to write the sum and just convert the last number (in string form) to int and add that with `+=` to our last total from our previous iteration.

If you do this correctly, which I do, you should get both the right formula, as string, and the right total, as int which we need to convert to string again for the `print()`.

The way I create an array of the string is like this:

```
input_to_array = [char for char in input_to_str]
```

Code Snippet

```
# Request an input
input = input()

# Convert the input to string
input_to_str = str(input)

# Now we can convert the string to an array
input_to_array = [char for char in input_to_str]

# Create a variable so we can get the result of the sum later on
sum = 0

# Create a variable so we can store the formula (1+2+3+4)
sum_str = ""

# We create a counter so we know at which position in the string we are
counter = 0

# Loop through the formula which we converted to an array
for c in input_to_array:
    # If length of the formula we pass is odd we make sure to add a "+"
and then the next number
    if len(sum_str) % 2:
        sum_str += "+" + input_to_array[counter]

        # If the length of the formula is anything but odd we only put the
next number in
    else:
        sum_str += input_to_array[counter]

    # We always add the next number to the last total of sum until
    sum += int(input_to_array[counter])

    # Move one up so we can actually get the next number
    counter += 1

# Now that we are done with the loop we print both the formula and the
sum as one whole string
print(sum_str + "=" + str(sum))
```

Assignment: A1W1P6 - Hours, minutes and seconds

Creation Date: 04-09-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

I wanted to have a variable for each calculation and since the input will be the amount of days I've casted that to an int and called the variable 'd' to calculate the hours.

Once you have the hours you can calculate the minutes and once you have your minutes you can calculate the seconds.

After the calculations are done I simply printed the outcome with the variables casted to a string so concatenation is possible.

Code Snippet

```
# Print what we need
print("Provide an amount of days")

# Request an input and convert it as int so we can use it in a formula
d = int(input())

# Calculate the hours, minutes and seconds with the input
hours = d * 24
minutes = hours * 60
seconds = minutes * 60

# Print the results as strings. Variables need to be string because we
otherwise concatenate an integer to a string
print("Hours: " + str(hours))
print(", Minutes: " + str(minutes))
print(", Seconds: " + str(seconds))
```

Assignment: A1W2A1 - Immediate successor

Creation Date: 06-09-2023

What did I learn?

How Codegrade handles errors - I also overlooked relatively simple solutions in favour of more difficult ones which could be written with fewer lines. This resulted in spending more time on this assignment than I wanted.

How did I learn it?

I've looked at the errors which were outputted by Codegrade and compared them to the errors I got from my IDE (PyCharm).

Why/how did I solve it?

This was an interesting one for me. Not that it was a difficult assessment but because I needed to think simpler than I'm used to.

This is why my first thought was using the `datetime` library to get the current date, automatically detect when a month did or didn't have 31 days ect.

Although I did get far with that library Codegrade didn't like the exceptions the library was causing on the website - the issue wasn't present on my laptop/pc.

Hence why I switched to my current version of the code (see below, under 'Code Snippet').

I used methods, why?: my main reason to use methods was to prevent that the code would become overwhelmingly long. A good example of this is the validation method, `validate()`. I could have changed the order of the code but since I was able to use methods I had the flexibility and therefore the possibility to validate the input more or somewhere else. So besides it being useful for fewer lines of code it is also very useful if you are trying to debug quick and easy.

Considering you used a library first, why would you choose a library over your current solution?:

Well, a library tends to be programmed in a way it already checks a lot of edge cases for you which gives you more information when you need to debug your software. The interesting part though is that this reason is also the reason why Codegrade didn't accept it. - The most likely reason is that Codegrade forced certain prints which your local machine doesn't force. And since `print()` methods are expected to be the actual result of the assignment it didn't work.

So how does it actually work?:

- **main()** triggers the whole program

- **dateInput()** first I'm asking for user input and pass that to the `dateInput()` method.

This method is responsible for triggering the validation and then seeing if the validation succeeded or not. If the validation succeeded and everything is OK it will trigger the shift functions and print the shifted outcome. This would complete the assessment.

- **validate()** the validate method checks if the values which are passed are correct and if not it prints an error and returns `None`, if it is all good it will actually format the date again to guarantee the year, month and day are in the correct place (`YYYY-MM-DD`) and return it.

- **shiftYear(), shiftMonth() and shiftDay()** are responsible that the year, month or day can be both increased or decreased.

Code Snippet

```
# Validate year format
def validate(date_text):
    year = date_text[0:4]
    month = date_text[5:7]
    day = date_text[8:10]
    if int(year.replace('-', '')) >= 2000:
        return f'{year}-{month}-{day}'
    elif int(year.replace('-', '')) < 1900:
        print("Input format ERROR. Correct Format: YYYY-MM-DD")
    else:
        print("What happened?")

# Function to increase or decrease (aka shift) the year
def shiftYear(date_text, shift):
    year = int(date_text[0:4]) + shift
    return str(year) + "-01-01"

# Function to increase or decrease (aka shift) the month
def shiftMonth(date_text, shift):
    month = int(date_text[5:7]) + shift
    return date_text[0:4] + "-" + str(month) + "-01"

# Function to increase or decrease (aka shift) the day
def shiftDay(date_text, shift):
    day = int(date_text[8:10]) + shift
    day_str = str(day)
    if len(day_str) < 2:
        day_str = "0" + day_str
    return date_text[0:7] + "-" + day_str

# Execute our actions here
def dateInput(input):
    i = validate(input)
    if i != None:
        i = shiftDay(i, 1)
        print("Next Date: " + i)
        i = shiftMonth(i, 1)
        print("Next Date: " + i)
        i = shiftYear(i, 1)
        print("Next Date: " + i)

# Get input and pass it to the dateInput() function
def main():
    date_input = input()
    dateInput(date_input)

# Run the program
main()
```

An example of using the datetime library

```
# Define the dateInput() method
def dateInput(input):
    # Trigger the validate method
    x = validate(input)

    # If the validation is OK
    if x != None:
        # make the validated format a date of type datetime
        datetime.date = x

        # To shift one day into the future replace the current day with
current_day+1
        datetime.date = datetime.date.replace(day=datetime.date.day +
1)

        # Print our new date
        print(datetime.date)

        # To shift one month into the future and begin on day one of
the month
        datetime.date = datetime.date.replace(month=datetime.date.month
+ 1, day=1)

        # Print our new date
        print(datetime.date)
```

Assignment: A1W2P1 - Even or Odd

Creation Date: 07-09-2023

What did I learn?

Nothing

How did I learn it?

Not applicable

Why/how did I solve it?

I requested user input, casted it to int and then did modulo (%) with 2 to see if the remainder is uneven.

If the remainder is not 0 then the result is uneven, if not the result is even.

Code Snippet

```
# Ask for input and immediately make it an int
i = int(input())

# Modulo to check if the input is even or odd as it will check the remainder
if i % 2:
    print("Odd")
else:
    print("Even")
```

Assignment: A1W2P2 - Leap year

Creation Date: 11-09-2023

What did I learn?

I learned how to calculate a leap year

How did I learn it?

I looked up the formula

Why/how did I solve it?

I requested user input and casted it to float. I've casted it to float because the outcome of the leap year can have decimals.

If we have the float we will divide it by 4 to see kickstart the calculation. Then if that result is odd it can't be a leap year, and if it is even it is (most likely) a leap year.

Code Snippet

```
# Get input
i = float(input())

# Divide by for
leap = i / 4

# If year is even after dividing it's a leap year, if not it is not a
Leap year
if leap % 2:
    print("not leap")
else:
    print("leap")
```

Assignment: A1W2P3 - Sides to shape

Creation Date: 11-09-2023

What did I learn?

I already knew it

How did I learn it?

I already knew it

Why/how did I solve it?

I requested user input and casted it to int. Once that was done I could use the `nr` in `shapes[]` as an index.

If that is done we can print `shapes[nr]` as `shape`.

Code Snippet

```
# List all kind of shapes
shapes = ["", "", "", "triangle", "quadrilateral", "pentagon",
"hexagon", "heptagon", "octagon", "nonagon", "decagon"]

# Get the number of the shape so we can index is later
nr = int(input())

# Request the index of the shapes as variable shape
shape = shapes[nr]

# Print shape
print(shape)
```

Assignment: A1W2P4 - Triangle type

Creation Date: 11-09-2023

What did I learn?

Nothing I already knew

How did I learn it?

Not applicable

Why/how did I solve it?

I requested user input and then checked through if and else if, if the sides are, or are not, equal to each other.

Based on if and how many sides are equal it will print what kind of triangle it is.

Code Snippet

```
i = input()

a = i[2]
b = i[7]
c = i[12]

if a == b and a == c:
    print("equilateral triangle")
elif (a == b and not a == c) or (a == c and not a == b):
    print("isosceles triangle")
elif (a != b) and (a != c):
    print("scalene triangle")
```

Assignment: A1W2P5 - Dutch holidays

Creation Date: 11-09-2023

What did I learn?

I didn't know `json.get(key)` was a built-in function in python.

How did I learn it?

I've looked up how python uses JSON and if I needed a library for it or not.

Why/how did I solve it?

I request user input, define a table of Dutch festivities with the `month and day` as key and the name of the festivity as value.

Then I use `festivities.get(input)` to see if there is a key based on the provided input. If the input is equal to an existing key it prints the name of the holiday.

If not, it will print "does not exist".

Code Snippet

```
i = input()

festivities = {
    "month 1, day 1": "Nieuwjaarsdag",
    "month 4, day 7": "Goede Vrijdag",
    "month 4, day 9": "Eerste Paasdag",
    "month 4, day 10": "Tweede Paasdag",
    "month 4, day 27": "Koningsdag",
    "month 5, day 5 ": "Bevreidingsdag",
    "month 5, day 18": "Hemelvaartsdag",
    "month 5, day 28": "Eerste Pinksterdag",
    "month 5, day 29": "Tweede Pinksterdag",
    "month 12, day 5": "Sinterklaas",
    "month 12, day 25": "Eerste Kerstdag",
    "month 12, day 26": "Tweede Kerstdag",
}

if festivities.get(i) is not None:
    print(festivities[i])
else:
    print("Does not exist")
```

Assignment: A1W2P6 - Dog years

Creation Date: 11-09-2023

What did I learn?

Nothing I already knew

How did I learn it?

Not applicable

Why/how did I solve it?

First of all I define (global) variables with one of them requesting user input. Then I check if the user input is a negative number. If it is lower than a 0 it will print an error and exit the program.

If it is a positive number it will enter a while loop and only exit once the iteration value (which is initially 0). In the loop we check if the iteration is smaller than 2 which, in other words, checks if the dog is younger than 2. If it is younger it will add 4 dog years, if it is either equal or bigger than 2 it will add 10.5 dog years. When the additions are done we'll increment the iteration value with `+= 1` so we won't stay in the loop forever. Once the loop is done we'll print the final age.

Code Snippet

```
i = int(input())
iteration = 0
age = 0
year = 0
dog_years_normal = 4
dog_years_first_two = 10.5

if i < 0:
    print("Only positive numbers are allowed")
    exit(1)
else:
    while iteration < i:
        if iteration < 2:
            age += dog_years_first_two
        else:
            age += dog_years_normal
        iteration += 1
    print(age)
```


Assignment: A1W2P7 - Chessboard colors

Creation Date: 11-09-2023

What did I learn?

I learned that `.lower()` and `.upper()` work within Python.

How did I learn it?

I already knew it existed in other programming languages, so I tried it in Python as well and it worked.

Why/how did I solve it?

I basically solved it by counting columns and rows which basically creates a grid and then because we invert colours each row we can use modulo to see if we have an even or uneven number. The same goes for the columns.

First I do check if the type of column is a string, and the type of row is an integer. This is how we can confirm that we correctly separated the algebraic notation.

When we did it correctly the column part will lower the whole alphabet string, loop through each position of the string and see if the index of the current iteration is equal to the column which we separated from the algebraic notation.

If the index is equal to the column we exit the loop, otherwise we add one to the column count, so we know we are not in the right column yet.

Almost the same goes for the rows, but this is a bit easier. In this case we enter a while loop instead of a for loop and we add one row (`r_count += 1`) as long as the row count is lower than the row (the number we separate from the algebraic notation).

When we did that we want to print either black or white, or white or black (since we need to invert the colours each time).

The way we could solve this is to see if the column count, but with an addition of one, is odd.

The reason that we add one is that we want to prevent to do modulo on a zero.

Now we did that we just simply check if the row is odd or not and then print the colour.

Now we are able to print black and white correctly based on algebraic notation.

Code Snippet

```
alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
c_count = 0
r_count = 0

i = input()
column = i[0].lower()
row = int(i[1])

if type(column) == str:
    for a in alphabet.lower():
        if a == column:
            break
        else:
            c_count += 1

if type(row) == int:
    while r_count < row:
        r_count += 1

if (c_count + 1) % 2:
    if (r_count % 2):
        print("black")
    else:
        print("white")
else:
    if (r_count % 2):
        print("white")
    else:
        print("black")
```

Assignment: A1W2P8 - Licence plate

Creation Date: 13-09-2023

What did I learn?

I've learned how `license.split()` and `.isnumeric` works in Python.

How did I learn it?

I looked up potential solutions to easily get the data from the license plate number, without retrieving the dashes.

Why/how did I solve it?

Since there is a whole list of patterns which needed to be marked as valid I began with a few patterns and seeing if my solutions worked. I constantly checked my changes as well, so I exactly knew when the script did or didn't know when a pattern was valid.

In the beginning, before I used methods, to mark if the license plate was correct I made use of a variable called `good` which would be set to `1` (true) so I could call the variable later and check if the license plate was in fact passing the validation.

The code would look something like this:

```
# Checking if the license plate is the length it should be
if license_plate_length == 8:
    # Are there dashes as 3rd and 6th character?
    if license_plate[2] == "-" and license_plate[5] == "-":
        # Are the first two characters the same as the 4th and 5th
        character?
        if (license_plate[0] + license_plate[1]) == (license_plate[3] +
license_plate[4]):
            # Are the last two characters NOT the same as the 4th and
5th character?
            if (license_plate[6] + license_plate[7]) !=
(license_plate[3] + license_plate[4]):
                # Then this license plater pattern is valid
                good = 1

# Print if the lincese plate is valid
if good:
    print("Valid")
else:
    print("Not Valid")
```

Eventually I switched to methods and made life a bit easier to validate license patterns. I think I've cut my code in half with using the `.split()` and `.isnumeric` method.

The `split("-")` method was a true lifesaver because it removes the dashes and makes a list for you. The dash basically says where the comma should go for the list. (So `XX-XX-99` becomes `[XX, XX, 99]`). When you can call them with an index and use `.isnumeric` you can also see if that stored string *only* contains numbers. This is really useful to because we

can now detect where the '99' is from 'XX-XX-99'. If you make variations on this with the `len()` method as well we suddenly can detect variations as 'XXX-99-X' as well. See the final code below.

Code Snippet

```
def validate_license(license):
    # Split into parts
    parts = license.split("-")

    # Validate format
    # Check if the license plate is long enough
    if len(parts) != 3:
        return False

    # We need to have a numeric value in our first and last 'part' if
    # there are any numerics
    if parts[2].isnumeric() != parts[0].isnumeric():
        return False

    # Checks for patterns:
    # XX-99-99 and 99-XX-XX
    if parts[1] == parts[2] and parts[0] != parts[1]:
        return True

    # Checks for patterns:
    # 99-99-XX and XX-XX-99
    if parts[0] == parts[1] and parts[0] != parts[2]:
        return True

    # Checks for patterns:
    # 99-XX-99 and XX-99-XX
    if parts[0] == parts[2] and parts[0] != parts[1]:
        return True

    # If every part is unique
    if parts[0] != parts[1] and parts[0] != parts[2]:
        # Checks for patterns:
        # 99-XXX-9, XX-999-X and XXX-99-X
        if (len(parts[0]) + len(parts[1])) == 5 and len(parts[2]) == 1:
            return True

        # Or check for (the reversed) patterns:
        # 9-XXX-99, X-999-XX, 9-XX-999
        if (len(parts[1]) + len(parts[2])) == 5 and len(parts[0]) == 1:
            return True

def _main_():
    # Request user input
```

```
license = input("License: ")

# Pass the user input to the method, so we can validate it and
immediately check if we get True or False back.
if validate_license(license):
    print("Valid")      # True
else:
    print("Invalid")    # False

# Run the script
_main_()
```

Assignment: A1W3A1 - Predefined templated

Creation Date: 17-09-2023

What did I learn?

Not applicable

How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it if I didn't learn anything new in the assignment/problem.

Code Snippet

Assignment: A1W3P1 - Simple palindrome

Creation Date: 18-09-2023

What did I learn?

I learned that you can join a list like this: `x = ''.join(myDict)`.

How did I learn it?

I've looked up easy and short solutions and wanted to do this with a list.

And, in my case, I needed to convert the list back to a string with join.

I didn't want to use something like this `x = mySeparator.join(myDict)` so I looked up a solution which is a tiny bit shorter and 'easier' to use.

I found and settled on `''.join(reversed_i)`. This is actually a really obvious solution as you replace the variable `mySeparator` with a string. So in core it basically works the same.

Why/how did I solve it?

The list is reached by casting the input (string) to a list like this:

```
i = input()
reversed_i = list(i)
```

And then reversing and stringifying the list like this:

```
reversed_i.reverse()
reversed_i = ''.join(reversed_i)
```

Code Snippet

```
# Ask for user input
i = input()

# characters to filter.
punct_marks = ",.?!;"

# Remove all characters listed in 'punct_marks'.
count = 0
for p in punct_marks:
    i = i.replace(p, "")
    count += 1

# Reverse filtered input
reversed_i = list(i)
reversed_i.reverse()
reversed_i = ''.join(reversed_i)

# Check if the input is the same as the reversed input
result = ""
result = f'{i} is '
if i == reversed_i:
    result += "a palindrome"
else:
    result += "not a palindrome"

# Print if the input is a palindrome or not
print(result)
```


Assignment: A1W3P2 - Advanced palindrome

Creation Date: 18-09-2023

What did I learn?

Not applicable as the code I used is basically the same as the code I made for [A1W3P1 - Simple palindrome](#).

The only difference is that the result print is expected to be different and I needed to add one extra `.replace()`. Specifically this replace `.replace(" ", "")` so it actually works with sentences.

How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it if I didn't learn anything new in the assignment/problem.

Code Snippet

Assignment: A1W3P3 - Modular rectangles

Creation Date: 18-09-2023

What did I learn?

Not applicable

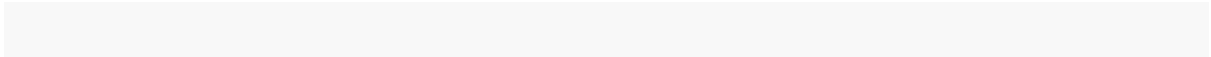
How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it if I didn't learn anything new in the assignment/problem.

Code Snippet



Assignment: A1W3P4 - Celcius to Fahrenheit

Creation Date: 18-09-2023

What did I learn?

Not applicable

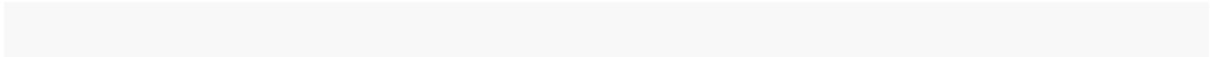
How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it if I didn't learn anything new in the assignment/problem.

Code Snippet



Assignment: A1W3P5 - Multiplication table

Creation Date: 20-09-2023

What did I learn?

I've learned what the enumerate function does and how I can use it in my code.

It returns a tuple with the first value as the count and the second one as the actual value from the list you're enumerating.

So if you use the following:

```
# Our List
list = ["item1", "item2", "item3"]

# Enumerating list and return a tuple
for count, item in enumerate(list):
    # Then print the count and the item
    print(count, item)
```

The output from the code above would be:

```
0 item1
1 item2
2 item3
```

The count will start at 0 and end when enumerate is finished enumerating through the list.

How did I learn it?

I've looked up easy and short solutions which I could use in this and future assignments.

Then I found what `enumerate()` does and how I could use it.

Why/how did I solve it?

I solved the assignment by making three different list and filling these with numbers by using a for loop with a specific range.

The `column(_joined)` variables are responsible for the right numbering while the rows actually does multiplication.

Then, at the end, once all variables are ready (so when all their respected values are added) we can enumerate on the rows (which we put in a list called `tables`).

Because of enumerate we can also count up with each print. This way we get the final output:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Code Snippet

```
# Define Lists
column = list()
row = list()
tables = list()
# Create columns
for n in range(0, 11):
    if n != 0:
        column.append(str(n))
    column.append("\t")
# Create rows
for n in range(1, 10):
    row.append(str(n) + "\n")
# Create tables
for i in range(0, 11):
    row = ""
    # Row 1
    if i == 0:
        for j in range(1, 11):
            row += str(j) + "\t"
    # ALL other rows
    else:
        for j in range(1, 11):
            row += str(j * i) + "\t"
    # Add the row
    tables.append(row)
# Join Lists so they both become a string
column_joined = "".join(column)
row_joined = "".join(row)
# Print each index (i) with their value (table) from the List (tables)
for i, table in enumerate(tables):
    if i < 1:
        print(f"\t{table}")
    else:
        print(f"{i}\t{table}")
```

Assignment: A1W3P6 - Binary to Decimal

Creation Date: 20-09-2023

What did I learn?

Not applicable

How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it if I didn't learn anything new in the assignment/problem.

Code Snippet

Assignment: A1W3P7 - Truth Tables

Creation Date: 20-09-2023

What did I learn?

Not applicable

How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it if I didn't learn anything new in the assignment/problem.

Code Snippet

Arch 2

[In Arch 2 I stopped doing problems, hence why this document is much shorter]

Assignment: A2W5A1 - Processing student data

Creation Date: 04-10-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it or what my code is if I didn't learn anything new in the assignment/problem.

Code Snippet

```
def validate_data(line):
    # Init basic variables
    corrupt = False
    invalid_data = []

    # Split the line we are given from the csv file
    studentnr, firstname, surname, birthdate, study = line.split(",")

    # Make a set of the special characters we want to filter out
    special_characters = set("!@#$%^&*()-+?_=<>/0123456789")

    # Check if our studentnr doesn't start on a '08' or '09'
    if not studentnr.startswith(("08", "09")):
        invalid_data.append(studentnr)
        corrupt = True

    # Check if we have special characters in our firstname
    if any(sc in special_characters for sc in firstname) or firstname == "":
        invalid_data.append(firstname)
        corrupt = True

    # Check if we have special characters in our surname
    if any(sc in special_characters for sc in surname) or surname == "":
        invalid_data.append(surname)
        corrupt = True

    if not birthdate[:4] or int(birthdate[:4]) not in range(1960, 2004):
        invalid_data.append(birthdate)
        corrupt = True

    # If a study is not provided
    if not study or study.upper() not in ["INF", "TINF", "CMD", "AI"]:
        invalid_data.append(study)
        corrupt = True

    # Now, after filtering, we put them in the correct list
    if corrupt:
        line = f"{line} => INVALID DATA: {invalid_data}"
        corrupt_lines.append(line)
    else:
        valid_lines.append(line)
```

Assignment: A2W6A1 - Addressbook

Creation Date: 10-10-2023

What did I learn?

I learned that you can use `dict.extend()` to merge multiple dictionaries.

How did I learn it?

I looked up how I could merge multiple dictionaries cause I knew it must be possible with a build in method.

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it or what my code is if I didn't learn anything new in the assignment/problem.

Code Snippet

```
def list_contacts(json, direction):
    # Check if we want to reverse the list (the same as
    # ascending/descending)
    reversed = (direction.upper() == "DESC")

    # Sort the contacts based on the boolean
    sorted_contacts = sorted(json, key=lambda x: x['id'],
                             reverse=reversed)

    # Make a list so we can fill the addressbook
    addressbook = []

    # Loop through the sorted contacts
    for i, contact in enumerate(sorted_contacts):
        # Store our new entry in an empty list
        entry = [str(i + 1), contact['first_name'],
                 contact['last_name'], contact['emails'], contact['phone_numbers']]
        addressbook.append(entry) # Add the entry to the addressbook,
        # which is now sorted

    # Return the sorted list
    return addressbook
```

Assignment: A2W7A1 - Name hasher

Creation Date: 10-10-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it or what my code is if I didn't learn anything new in the assignment/problem.

Code Snippet

```
def encode_string(data: str, key: str = None) -> str:
    # a@b.c>d#eA
    # Check if data is a string
    if not isinstance(data, str):
        raise TypeError("data must be a string")

    # Check if key is a string
    if not isinstance(key, str):
        raise TypeError("key must be a string")

    # Replace single quotes with double quotes so the json can be
    # parsed correctly
    key = key.replace("'", '"')

    # Parse string as JSON directly
    key = json.loads(key)

    # Verify if the key was converted to dict
    if not isinstance(key, dict):
        raise ValueError("Unable to convert key to JSON")

    # 'Encode' the data
    for letter in data:
        encoded_values.append(key.get(letter))

    # Return the encoded string
    return ''.join(encoded_values)
```

Arch 3

Assignment: A3W09A1 - Car parking

Creation Date: 07-11-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

With the car parking assessment I created a 'quick-test.py' file so I could easily check if the code what was being run by codegrade was passing in the IDE. If it was then I pretty much knew Codegrade would accept it as well. (Although there were instances where the behaviour was different)

Code Snippet

```
# This is an example of 'quick-test.py' which would verify is Codegrade would mark it as a success
import carparking as cp
from datetime import datetime, timedelta

cpm = cp.CarParkingMachine(capacity=2, hourly_rate=4.0)
cpm.check_in("BB-494-H")
cpm.check_in("HH-494-B", datetime.now() - timedelta(hours=2))
print(cpm.get_parking_fee("HH-494-B"))
```

Assignment: A3W10A1 - Car parking logger

Creation Date: 11-14-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

With the car parking assessment I created a 'quick-test.py' file so I could easily check if the code what was being run by codegrade was passing in the IDE. If it was then I pretty much knew Codegrade would accept it as well. (Although there were instances where the behaviour was different)

Code Snippet

```
def write_to_log_file(self, new_line: str):  
    # open the log file  
    log_file = self.open_log_file()  
  
    # use seek to start writing at the beginning of the file and write the new line  
    log_file.seek(0)  
    log_file.write(new_line)  
  
    # truncate the file to the current position and close the file  
    log_file.truncate()  
    log_file.close()  
  
    return True
```

Assignment: A3W11A1 - Car parking extended

Creation Date: 11-14-2023

What did I learn?

I learned nothing with this assignment.

How did I learn it?

Not applicable

Why/how did I solve it?

With the car parking assessment I created a 'quick-test.py' file so I could easily check if the code what was being run by codegrade was passing in the IDE. If it was then I pretty much knew Codegrade would accept it as well. (Although there were instances where the behaviour was different)

Code Snippet

Arch 4

Assignment: A4W13A1 - Car parking final

Creation Date: 11-14-2023

What did I learn?

I learned that you need to commit your changes with sqlite3 otherwise it won't properly update the database.

How did I learn it?

Mostly through trial and error and a bit of research.

Why/how did I solve it?

With the car parking assessment I created a 'quick-test.py' file, so I could easily check if the code that was being run by codegrade was passing in the IDE. If it was then I pretty much knew Codegrade would accept it as well. (Although there were instances where the behaviour was different)

Code Snippet

```
# Drop and create database for a new run using __init__
def __init__(self, id: chr, capacity: int = 10, hourly_rate: float =
2.50, parked_cars=None):
    """
    :param id: The unique identifier for the CarParkingMachine. It
    should be a single character.
    :param capacity: The maximum number of cars that can be parked in
    the CarParkingMachine. Default value is 10.
    :param hourly_rate: The hourly rate of parking in the
    CarParkingMachine. Default value is 2.50.
    :param parked_cars: A dictionary that stores the parked cars and
    their corresponding parking details.
                        If not provided, an empty dictionary will be
    used.
    """
    # check if parked_cars is None and set it to an empty dict if it is
    if parked_cars is None:
        parked_cars = {}

    self.id = id
    self.capacity = capacity
    self.hourly_rate = hourly_rate
    self.parked_cars = parked_cars
    self.parked_cars_stored = []
    self.logger = CarParkingLogger(id)
    self.json_file = f"{id}_state.json"
    self.db_conn = sqlite3.connect(os.path.join(sys.path[0],
'carparkingmachine.db'))
    self.db_conn.execute('''DROP TABLE IF EXISTS parkings;''')
    self.db_conn.execute(
        '''CREATE TABLE IF NOT EXISTS parkings (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            car_parking_machine TEXT NOT NULL,
            license_plate TEXT NOT NULL,
            check_in TEXT NOT NULL,
            check_out TEXT DEFAULT NULL,
            parking_fee NUMERIC DEFAULT 0
        );'''
    )
    self.check_table_exists(self.db_conn, "parkings")
```


Assignment: A4W14A1 - Name hasher 2

Creation Date: 11-14-2023

What did I learn?

I didn't really learn anything as this assessment was almost a copy and paste from the first name hasher assessment.

How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it or what my code is if I didn't learn anything new in the assignment/problem.

Code Snippet

```
def validate_values(encoded: str, decoded: str, key: str = None) ->
bool:
    if key is None:
        key = "A%B&C(D)E*F+G-H/I0J<K=L1M!N90?P>Q7R#S5T;U:V[W]X~Y$Z@"

    # Check if encoded and decoded are both strings
    if not all(isinstance(item, str) for item in (encoded, decoded)):
        raise TypeError("Both `encoded` and `decoded` must be strings")

    # If key is provided, check if it's a string
    if key and not isinstance(key, str):
        raise TypeError("`key` must be a string")

    # Use the decoding function
    resolved = decode_string(encoded, key)

    # Return True if the decoded value matches the given decoded
    string, False otherwise
    return resolved == decoded
```

Assignment: A4W15A1 - Folder structure

Creation Date: 12-22-2023

What did I learn?

I didn't really learn anything as this assessment was almost a copy and paste from the first name hasher assessment.

How did I learn it?

Not applicable

Why/how did I solve it?

As Cigdem Okuyucu said this to me; I won't specify how I solved it or what my code is if I didn't learn anything new in the assignment/problem.

Code Snippet

```
def rec_print_folders(n: int, pref: str, root: list) -> None:
    """
    This function prints the contents of a given root folder with
    indentations.
    """
    for i in root:
        if type(i) is list:
            rec_print_folders(n + 1, pref + '>', i)
        else:
            print(pref + '-' + i)

def rec_count_files(root: list) -> int:
    """
    The functions counts number of files in a given folder (and all its
    sub-folders).
    :param root: A nested list: an element either is a file (name) or a
    list as a sub-folder.
    :return: The number of files in the given folder
    """
    if len(root) == 0:
        return 0
    else:
        count = 0
        for i in root:
            if type(i) is list:
                count += rec_count_files(i)
            else:
                count += 1
        return count
```

3.2 Overview of activities outside Basecamp

One of my jobs is as a student researcher for Chemical Technologies at RDM kade (Hogeschool Rotterdam).

This job began as "Project 5/6" which was part of Technical Computer Science. The project team existed out of 4 students, including myself. One of whom was a friend of mine, Niko van Ommen.

A brief summary of what Chemical Technologies does:

One of the teams at Chemical Technologies develops and improves a mix of chemicals which is based on certain fruit sugars. The chemical changes colour if there is a certain chemical nearby, such as Ammonia. They added a colour identifier so you can see if the chemicals react at the speed they should react at, and if they go back to their original colour. The colour identifier also proved to be useful for additional development (see "A brief summary of what Niko van Ommen and I do").

A brief summary of what Niko van Ommen and I do:

During Project 5/6 the project team, including myself, created a relatively simple colour scanner for Chemical Technologies with some basic software.

Niko and I now, after Project 5/6, further develop both hardware and software which links the colour change to a 'reaction intensity' value. The software includes an android app which is created in Flutter and the software which runs on the hardware is written in C++.

The hardware setup is currently as follows:

- The Firebeetle ESP32e (Microcontroller)
- The Adafruit TCS34725 (RGB Scanner)

We plan to replace the RGB scanner with a spectral color sensor which should give more accurate colour readings.

The goal:

Our current goal is to finish and publish a good research paper, together with Chemical Technologies, for IEEE. This goal is set by Chemical Technologies so we (as in Chemical Technologies, Niko and myself) can prove the solution we are developing is an improvement on what currently exists, and that it can be used in a more day-to-day environment instead of a controlled environment such as a laboratory.

What do I do?:

I'm currently rewriting the mobile app because there were a few technical issues in the app which now seem to be resolved in the rewrite. The rewrite also included changes to the user interface as there were too many menus which seemed to cause confusion to our testers and our product owner.

Niko and I specifically chose for both Java and Dart as this is easy to set up, program with and it has direct syntax support in our IDE of choice, Android Studio.

The rewrite does not yet support automated testing and bluetooth but we will add this in the future.

During the rewrite I also focus on a possible iOS implementation, meaning that the app would be able to run on Apple devices such as iPhones. Although this is an 'extra' we currently want to keep this in our list as a lot of people also don't have an Android phone. If we would want to use the app in a government setting for example, they would most likely

have an iPhone from their work. (I know this out of experience as I worked at the Palace of Justice for a year, in the 'on location' IT support team from "Regio 5".

New functions of the app include being able to dynamically load new pages by using the navigation bar and page objects. A working code example is:

```
class _MyHomePageState extends State<MyHomePage> {
  // Start app on home page (1) and keep tabs on the current page
  int currentPageIndex = 1;
  // Select the page content with switch cases
  Widget _getPage(int page) {
    switch (page) {
      case 0:
        return const ConnectPage();
      case 1:
        return const HomePage();
      case 2:
        return const InfoPage();
      default:
        return const HomePage();
    }
  }

  @override
  // Build the navigation bar with the logic to switch pages if the correct button is
  // pressed
  Widget build(BuildContext context) {
    return Scaffold(
      bottomNavigationBar: NavigationBar(
        // On navigation select
        onDestinationSelected: (int index) {
          // Set correct page by index
          setState(() {
            currentPageIndex = index;
          });
        },
        // Give the navigation a specific style
        backgroundColor: HexColor("#EBEBEB"),
        indicatorColor: HexColor("#D8DBE2"),
        // Highlight the button from the current page
        selectedIndex: currentPageIndex,
        // Add the buttons which we need to press to change pages
        destinations: const <Widget>[
          NavigationDestination(
            selectedIcon: Icon(Icons.add_circle),
            icon: Icon(Icons.add_circle_outline),
            label: 'Connect',
          ),
          NavigationDestination(
            selectedIcon: Icon(Icons.home),
            icon: Icon(Icons.home_outlined),
            label: 'Home',
          ),
          NavigationDestination(
            selectedIcon: Icon(Icons.info),
            icon: Icon(Icons.info_outline),
            label: 'Info',
          ),
        ],
      ),
      // Set the new page content
      body: _getPage(currentPageIndex),
    );
  }
}
```

```
}  
}
```

Learning Kubernetes

Author: Ruben Flinterman

Before I begin, these are the terms I'll be using in my text and what they mean:

- **GitHub Actions:** GitHub Actions allows you to automate tasks like building, testing, and deploying your code directly from your GitHub repository. Workflows are defined in YAML files within the `.github/workflows` directory of your repository.
- **Workflow:** A GitHub workflow is a collection of jobs that run on GitHub-hosted runners or your own self-hosted runners. It can be triggered by various events such as pushes, pull requests, or scheduled intervals.
- **Kubernetes:** A powerful system for managing containerized applications. It helps automate the deployment, scaling, and management of containerized applications.
- **Image:** A lightweight, stand-alone, and executable software package that includes everything needed to run a piece of software. It serves as a blueprint for creating containers, defining the application, its dependencies, and other settings. Images are used to package and distribute applications in a consistent and reproducible manner.
- **Container:** A lightweight, standalone, and executable software package that includes everything needed to run a piece of software, including the code, runtime, libraries, and system tools. Containers provide a consistent and efficient way to deploy and run applications across different computing environments.
- **Pod:** A pod is like a group of friends. It's one or more containers (like software packages) that work together and are placed on the same computer.
- **Deployment:** Think of deployment as a set of instructions on how to manage a group of these containers (pods). It helps in organizing and running them.
- **Service:** Imagine a service as a waiter in a restaurant. It's like a helpful link that lets different pods (containers) talk to each other easily.
- **Ingress:** Ingress is like the entrance to a building. It helps manage how external things (like internet traffic) get access to the services inside our group of containers.

As a hobby I've been researching and trying out Kubernetes and since last weekend (9-12/10-12) I started to use my Kubernetes cluster as a production environment. And I've been recently hired at the Datalab to make scaling possible there too.

The idea of Kubernetes is that I am able to deploy web applications in the form of container images on multiple servers at once, if required, so I can both scale horizontally (more servers) and vertically (upgrading servers or more pods) to prevent services from going offline, unless I tell it to go offline. The services being online are important as I'm hosting both my own and also my father's website who has a small company (self-employed next to his main job) in production. Although traffic is still minimal I expect it to grow overtime. Migrating took time and if I did this when traffic was higher it could have done some damage as potential investors wouldn't be able to reach his website.

Next to that it's now easier to deploy applications and cut down costs too. Before I was paying for both DirectAdmin and a VPS to run it with nginx as back-end. Now instead of that, I'm using the nginx ingress controller for kubernetes which potentially cuts costs of around 10 euros a month. And regarding deploying code, I am now able to push my code to Github what builds a container image using a Github Action/Workflow. When that image is published privately (linked to the repository I pushed to) I can pull it on Kubernetes by restarting the "deployment" (which happens to include it's own "service"). This then easily hooks into my ingress controller using an "ingress" which makes everyone able to call a "pod" using a domain name. Both www.subdomain.example.com and www.example.com would be a valid domain for the ingress.

So although kubernetes is not a programming language it does provide a few things regarding experience which could be important for a developer (with an eye for writing safe software):

- 1) How you can both easily and safely configure services, so it won't give any issues on long term (IP, ports, subnets)
- 2) It gives you a general understanding of networking
- 3) It gives you a general understanding of yaml and bash scripts
- 4) It forces you to properly think-out your infrastructure. E.g.: Do you want a database next to your web application? Do they need to run on the same cluster? And if so, do they need to be in the same pod?

Class based rewrite of Challenge Week 1 in my spare time

In some of my spare time a friend (from Technical Computer Science) and I challenged each other to quickly come up with a game concept while still following the guidelines of challenge 1.

As we are both pretty experienced with writing software (we did projects with each-other before) we did add some rules of our own.

The additional rules:

- AI is not allowed unless you use it for JSON.
- It can't have errors.
- You can't copy any existing work (Websites like stackoverflow are allowed)
- You can use libraries as long as it's build into Python.

The normal rules:

Minimal requirements:

- 10 locations
- 10 items
- 5 puzzles
- All python techniques of week 1-3
- No errors
- No copies of work from fellow students or adventures found on the internet
- Theme must be SPACE TRAVEL

Nice to have:

- Images (look at the turtle library in python)
- Sounds (depends on your operating system)
- Understand commands like GO WEST, OPEN THE DOOR, PICK UP LASER

Obviously, for a challenge for a small evening we didn't do 10 locations, 10 items or 5 puzzles. This much wouldn't be reasonable in a few hours.

After the challenge:

This resulted into a fun challenge from around 2 hours after which we stopped (we both didn't finish).

Why we thought it was fun is because even though we had a lot of rules we both came up with a whole other concept.

He went for a 'Super Mario Bros. (1985)' styled game but then based completely on ASCII.

I went with the game concept we used during the challenge week, so I could look how much time I would need to write a game

with the same core concepts (so ignoring stuff like a story). So a 'Space Quest I' like command system but without any real visuals.

Some of the screenshots we shared early in the evening were:

His Super Mario Bros like game:

[illegible]

My game, a Space Quest like game:

```
=====
You start in the game room.
Description: A spacious game room with various games and entertainment.
=====
What do you want to do?
```

You can find my code on GitHub: <https://github.com/RFlintstone/text-based-adventure-python/blob/main/main.py>

I made `World` a parent class, `Player` is a child of `World` as it is 'standing' in the world, and then we also have a class called `Console`.

World: Responsible for changing rooms

Player: Responsible for the player actions and inventory

Console: Responsible for the screen width so we have some sort of implementation for a resolution.

Some of my code is located on the next page.


```

# Our Classes
c = Console()
w = World()

# Prettify the console
print("=" * c.get_screen())

# Ask for input
p = Player(input("What's your name? "))

# Add dummy item so we can test the inventory
p.add_inv("key")

# Set our first room
w.set_room("game room")

# Prettify the console
print("=" * c.get_screen())

# Give our first starting location
print(f"You start in the {w.get_room()}.")
print(f"Description: {w.load_room()}")

while True:
    # Every time we will be asked a command we also prettify the
    console
    print("=" * c.get_screen())
    cmd, arg = p.command(input("What do you want to do? "))

    # Choose an action based on the tuple from 'p.command'
    if cmd == "go":
        # Go to a different room, if it exists
        p.move(arg)
    elif cmd == "use":
        # Use an item, the item would be removed from the inventory
        p.use(arg)
    elif cmd == "inspect":
        # Inspect things from a room
        res = p.inspect(arg)
        # Then print the result of our inspection
        print(res)
    elif cmd == "talk":
        # Talk to someone - I ignored this in the 1v1 challenge because
        of time constraints
        p.talk(arg)

```