

Intro to Creative Computing

Week 11: Repetition!

Tasks

- **Written on paper**
 - Not marked
 - But checked
-
- Task 1: Print your name!

Tasks

- Task 2: Using a **for loop** count to 5 ?
- Task 3: How would you rewrite it to count from 3 to 23 ?

Tasks

- Task 4: Write a function which checks if a number is odd
- *Hint: Return True or False depending on if it's odd*
- *Hint: You can use modulo (sign %)*

Tasks

- Task 5: Write a function which selects a maximal number from a list, write the checking yourself
- *Hint: Don't forget to return the number*
- *Bonus: Check if your function actually got a list with more than just 1 element in it!*

Tasks

- Task 6: Make a dictionary to translate 3 selected English words to Spanish. Use this dictionary to translate a word input by user – and if you don't have this word in the dictionary print "UNK"
- *Potato = Patata*
- *Cat = Gato*
- *Dog = Perro*

Tasks

- Task 7: Load a text from a file and print it's lengths (the number of characters)

Tasks

- Task 8: Convert a string to **lower case**, **strip it's spaces** (empty characters like " ") and then give me the substring from the 5th to 10th place
- *Hint: Indexing strings works like indexing lists!*

Tasks

- Task 9: Extract information from a string:
 - String = "model_potato_3layers_128units_griffLim"
- Extract the name of the model ("potato")
- *Hint: You can split the string into list by for example "_" then you would be able to get the name easily*
- Bonus: Extract also the number of layers (3) and units (128)

Tasks

- Task 10: Check if a string contains a word “Alice”

Hint: print the result (saying “yes, it contains” or “no, it doesn’t”)

Tasks

- Task 11: Check if a string is exactly “Alice”
- Task 12: Check how many times the string contains “Alice”

Tasks (last one!)

- Task 13: Write a class for point
- *Hint: point should have x,y*
- *Hint: as a method it should be able to give you distance from <0,0>*

Pause 1

Assignment

- Write a small two-player text game where the first player enters a list of 5 things or people they love most, and the second player tries to guess what those things are.

1: Greet the players and prints a description of how to play.

2: Prompt user input for 5 different things or people, then receive that input and saves it. Input each item separately. Then ask the player to input a clue for each item and saves the clue so that it corresponds to the item. Try figuring out how to hide the player's input in your console, or just print a bunch of spaces to hide it!

3: Offer clues to the second player and prompt the player to guess! Go one item at a time. Print out the associated clue and prompt the user to guess the item. Keep a score for how many times the player guesses wrong for each item. When a player gets an item, print a congratulations message. At the end of the whole list, start back at the beginning until all of the items are guessed. If an item has already been guessed correctly, skip that item and move on to the next. Do this for however many turns it takes for the player to guess all of the items. At the end of the game, print the score. Save the score to a file as the best high score.

4: Ask the player if they'd like to replay. If so tell them the all time best high score and replay the game. If not quit the game and tell the player their best score before exiting.

- Due to 10.1.2020 => mail to: v.ruzicka@arts.ac.uk

Machine learning

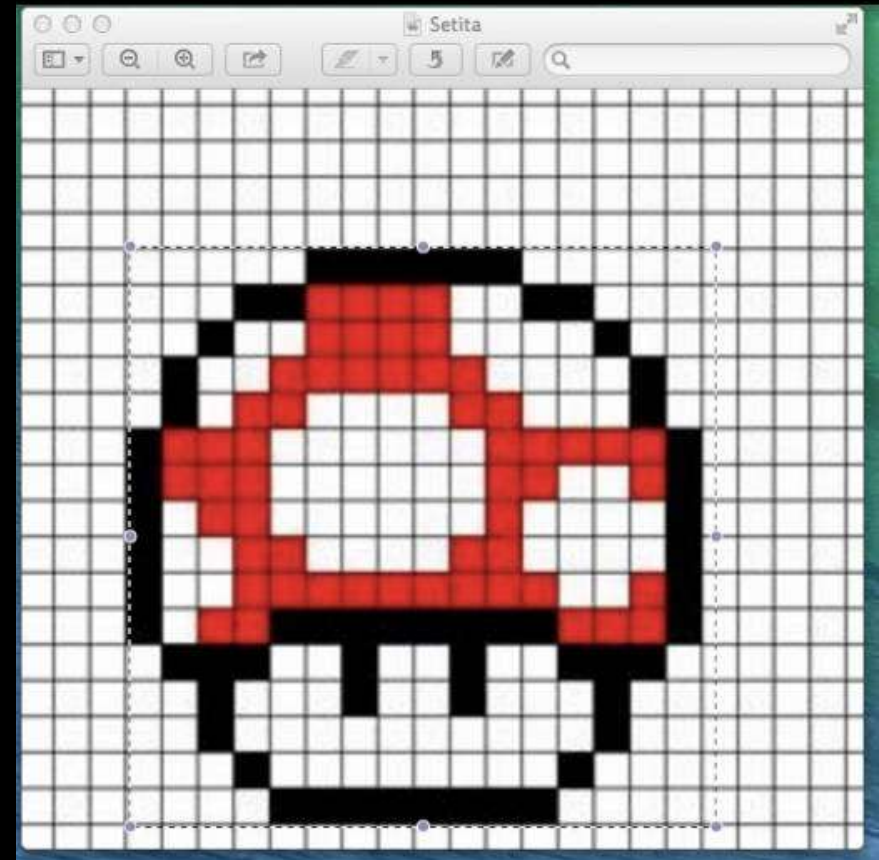
- **Good reason to know python** – most of Machine Learning is using it!
 - Academic research = laboratories at Universities
 - Sometimes theoretical (sort of like philosophy of arts)
 - Sometimes applied – using these tools suggested by ^ to one specific task
- Some of them are applied on arts / on creative AI / on human-computer interaction

Machine learning

- Can be understood as a black box ...
 - Which has **some inputs** on one side ...
 - And **some outputs** on the other side ...
 - (In this way it can be thought of as a “box” in a processing pipeline)
-
- For example:
 - Inputs: 512 random numbers from -1 to 1
 - Outputs: generated images

What are images?

- How computer sees images? As information
- As a grid of values of RGB / Grayscale:

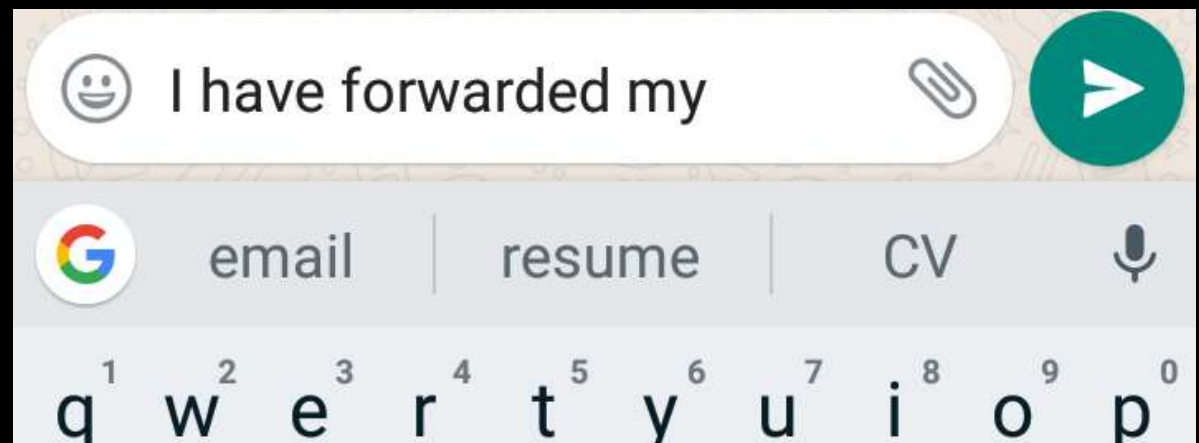


Example 1: Generating faces

- Cool thing: Trying out a generative model! (and the pains of setting up Colab sheets)
- Generative Adversarial Networks (GANs)
 - First watch: <https://www.youtube.com/watch?v=XOxxPcy5Gr4>
 - Play with it: **DEMO 1** = https://colab.research.google.com/drive/1Vq_rPfTp6lvOcaO4o4d4klyBzk6xTEpl

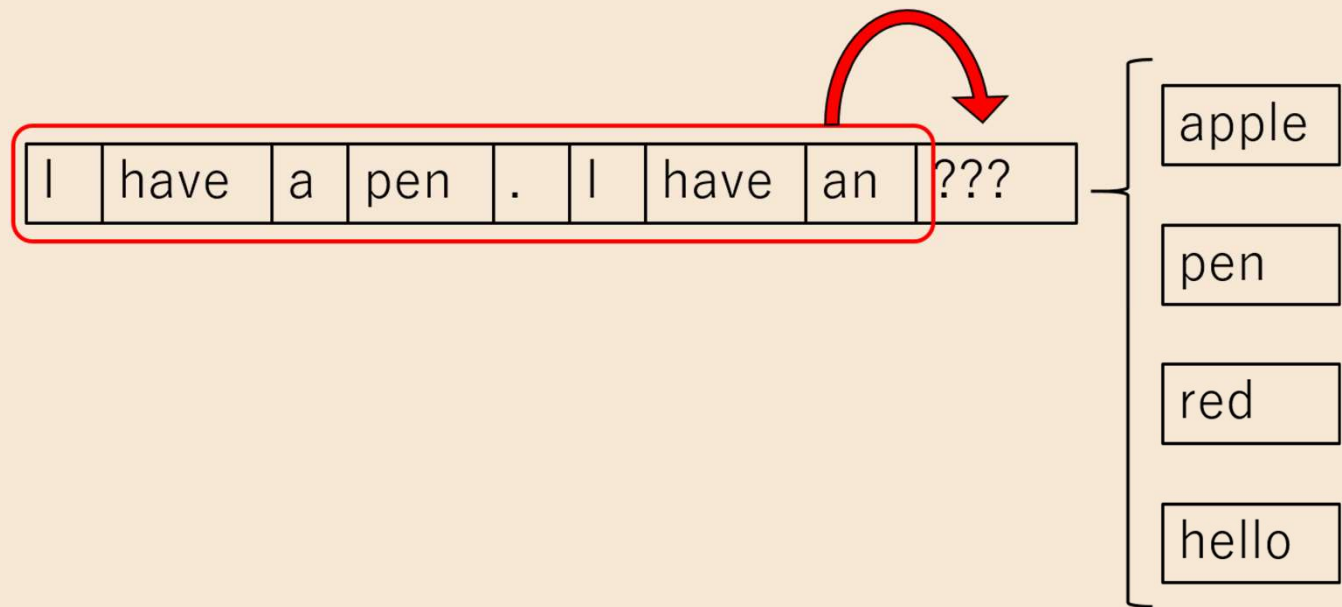
Example 2: Generating text

- Predicting next words?



Example 2: Generating text

- Predicting next words?



Example 2: Generating text

- GPT-2 is a model to complete sentences ...
- Model prompt >>> My favorite genre of music is
- ===== SAMPLE 1
jazz. It's not a genre I grew up in, and yet I've played, sung and performed a lot of it. But for me, jazz is the musical equivalent of my inner child. I love jazz because it's a perfect match for my strengths: intelligence, creativity, and a drive to do whatever it takes to achieve my artistic ideals. My dream job involves doing what I love. I want to be a composer, novelist, playwright, musician, or dancer. I believe I have the ability to do this, and I'm ready to take on the challenges of being a successful entrepreneur.
- ===== SAMPLE 2
hip-hop. The people who I hangout with are the most interesting, the most interesting people. We love to listen to the latest hip-hop. A lot of the artists are really creative and crazy. You always find a new thing every day. What's the best place to grab a beer?

Example 2: Generating text

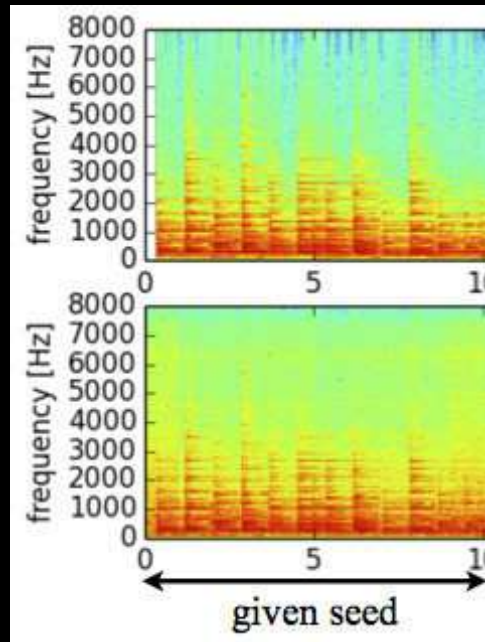
- GPT-2 is a model to complete sentences ...
- Play around with GPT-2 here:
 - **DEMO 2** =
https://colab.research.google.com/drive/1xmCwjKZmBqJ5ala_6ZVHNZwPRY_UdSSm

Example 3: Generating audio

- Audio is like text, right? ...
- Not really ... no alphabet (no limited amount of signs), really long (many samples, when compared to text sentences)
- Intermediate representation: Spectrogram (image to represent audio)

Example 3: Generating audio

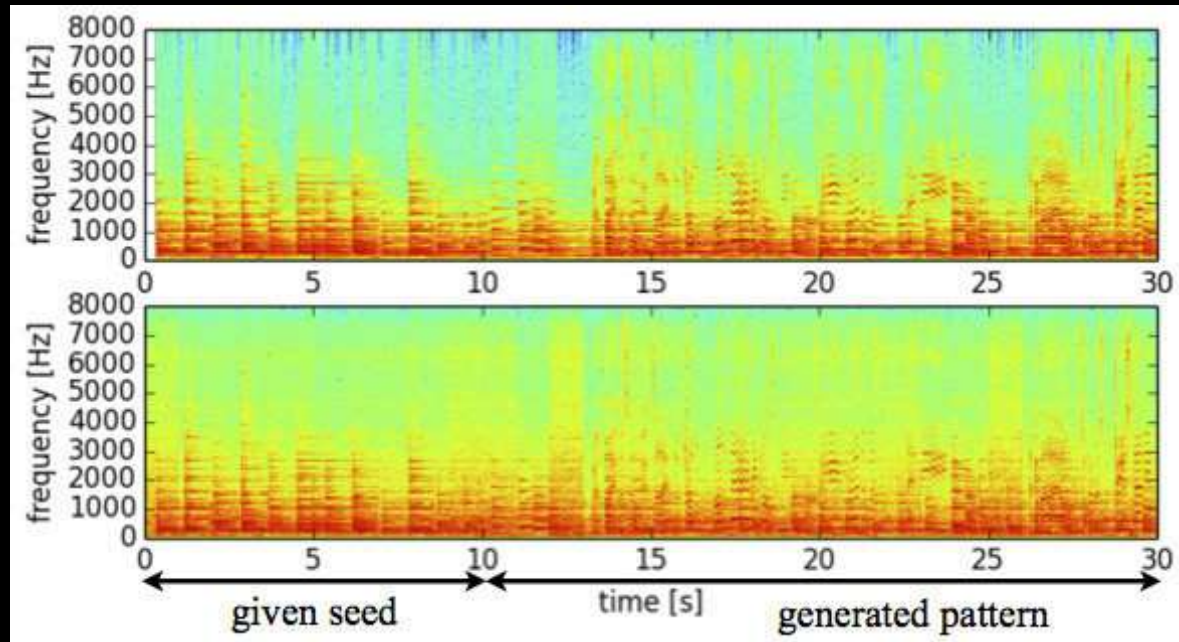
- Intermediate representation: Spectrogram (image to represent audio)



???

Example 3: Generating audio

- Intermediate representation: Spectrogram (image to represent audio)



Example 3: Generating audio

- **Demo 3:**
 - Input: *Jazz* <https://www.youtube.com/watch?v= MOpX3R7CwY>
 - Generated: <https://soundcloud.com/previtus/ml-jazz-meanderings-ml-generated-sounds-1/s-DCZbx>

Motivation?

- Motivation for these demos was to show that Python can be quite powerful ...
- ... and that you all are now equipped to use these already existing samples of code ...
- And that's quite often what you need to do – start with an example, tinker with it to change what it does (*for example train it on your own data, or interact with it in some fun way ...*)
- Writing your own projects will come in time too ...

Pause 2

Libraries

- What is a library?



Libraries

- What is a library?

```
import os
import math
import sys
import matplotlib

print(math.pi)
print(sys.version)
```

Libraries

- What is a library?
- *(semi-definition) Python library is a collection of functions and methods that allows you to perform many actions without writing your code.*
 - Often community made
 - Often ready made for fast work
 - Often with cool examples!

Versions?

- python, pip,
python2, pip2,
python3, pip3

- conda

```
import sys  
print(sys.version)
```

```
# linux + mac  
  
which python  
python --version  
  
which pip  
  
conda info
```

```
# windows  
  
where python*  
python --version  
  
where pip  
  
conda info
```


Versions?

- conda

```
conda info
```

- *Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.*

Path?

- Example (my pc without python):
- `C:\Users\vitek-ntb-win>echo %PATH%`
- `C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;D:\Program Files\CalibreEPUB\;C:\Program Files\PuTTY\;D:\Program Files\Git\;C:\Users\vitek-ntb-win\AppData\Local\Microsoft\WindowsApps;C:\Users\vitek-ntb-win\AppData\Local\GitHubDesktop\bin`

```
# linux + mac  
echo $PATH
```

```
# windows  
echo %PATH%
```

Libraries ... ?

- How does this have anything to do with libraries?

```
import matplotlib
```

- Besides some default libraries which come in all python versions ([standard libraries](#)), many other require installation and also have versions – you might need to do this first:

```
pip install matplotlib
```

... actually

```
pip install matplotlib
```

- This ^^ only works in a case where someone has released a built code for us
- If it's a small and obscure code, we might need to

```
pip install git+https://github.com/matplotlib/matplotlib.git
```

... actually

```
pip install matplotlib
```

- This ^^ can also sometimes install a new version of the code, while the rest of our code relies on some older version, in that case ...

```
pip install matplotlib==
```

```
pip install matplotlib==2.0.2
```

... actually

- All of this is **usually relatively doable** ... you just test if it works and if it doesn't *someone online probably encountered the same problem* and their solution can help ...

The task?

- Find a python library of interest
- Install it
- Try it's examples
- And show it to me!

The task?

- Find a python library of interest
- Install it
- Try it's examples
- And show it to me!
- Some places to start?
 - <https://medium.mybridge.co/34-amazing-python-open-source-libraries-for-the-past-year-v-2019-93d6ee11aceb>
 - <https://github.com/vinta/awesome-python>
 - Bokeh (nice graphs), Pyxel and Pygame (games), BeautifulSoup and nltk (natural language processing)

The task?

- Find a python library of interest
- Install it
- Try it's examples
- And show it to me!

```
# hint:  
# if things don't work on local, use  
# google colab and this
```

```
!pip install matplotlib
```

- Some places to start?
 - <https://medium.mybridge.co/34-amazing-python-open-source-libraries-for-the-past-year-v-2019-93d6ee11aceb>
 - <https://github.com/vinta/awesome-python>
 - Bokeh (nice graphs), Pyxel and Pygame (games), BeautifulSoup and nltk (natural language processing)

Intro to images

- How computer sees images?

Images

- How can we load an image?
- How can we alter an image?
- And then save it?