

Syntax and Variables

BILD 62

Objectives for today

- Confirm rules of Python syntax
- Define three types of variables: integers, floats, and strings
- Concatenate & slice strings
- Determine rules for variable names

There are different types of programming languages, each with their own syntax, or rules.

- **Syntax:** the rules of a programming language
 - Includes punctuation, spacing, indentation, etc.
- Each language has strengths & weaknesses.
- Regardless, each language ultimately needs to communicate with the hardware of the computer, in 1's and 0's.
 - It's similar to DNA! And similar to DNA, we don't often describe it in individual base pairs. Instead we describe genes and describe DNA in a higher level way.

Syntax Rules in Python

- Python is **case sensitive**: `letsroll` \neq `LetsRoll`
- White space does not matter (e.g., line 9 or 11 below)
- Indentation *does* matter — use **tab** to indent your code
- **Indexing** (we'll come back to this later)

- Python starts indexing at 0. So if you have a list of numbers:

```
list = [ 2 , 5 , 7 , 1 , 9 , 2 ]
```

and you ask for `list[1]`, you'll get 5.

- Code that follows `#` is not read by the interpreter:

```
8 # this is a commented line!  
9  
10 print('this line will totally run.')  
11
```

Creating new variables

- Names are always on the left of the `=`, values are always on the right
- Pick names that describe the data / value that they store
- Make variable names as **descriptive** and **concise** as possible (this is an art!)
- Variables cannot be Python keywords:

```
[>>> import keyword  
[>>> print(keyword.kwlist)  
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def',  
 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lamb  
da', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']  
>>> █
```

(There are other rules for variable names....)

Python has many variable types, and each function a little bit differently.

Understanding your variable type is crucial for working with it.



Built-in simple variable types in Python

Type	Example	Description
<code>int</code>	<code>x = 1</code>	integers (i.e., whole numbers)
<code>float</code>	<code>x = 1.0</code>	floating-point numbers (i.e., real numbers)
<code>complex</code>	<code>x = 1 + 2j</code>	Complex numbers (i.e., numbers with real and imaginary part)
<code>bool</code>	<code>x = True</code>	Boolean: True/False values
<code>str</code>	<code>x = 'abc'</code>	String: characters or text
<code>NoneType</code>	<code>x = None</code>	Special object indicating nulls

Integers, strings, floats

function to convert to integer



- **Integers** (**int**): any whole number
- **Float** (**float**): any number with a decimal point (floating point number)
- **String** (**str**): letters, numbers, symbols, spaces
 - Represented by matching beginning & ending quotes
 - Quotes can be single or double; use single *within* double
 - Use \ to ignore single quote
 - Concatenate strings with +

Checking variable types

This is a very useful troubleshooting step!

- You can check what type your variable (**a**) is by using **type(a)**

- Alternatively, we can use:

```
>>> type(a) is float
```

or

```
>>> isinstance(x, float)
```

- Python lets you change the type of variables, however, ***you cannot combine types.***
- Use **del** to delete variables

It's important to know the precision of your variables.

In most datasets, we are working with floats.



Autopsy Report:

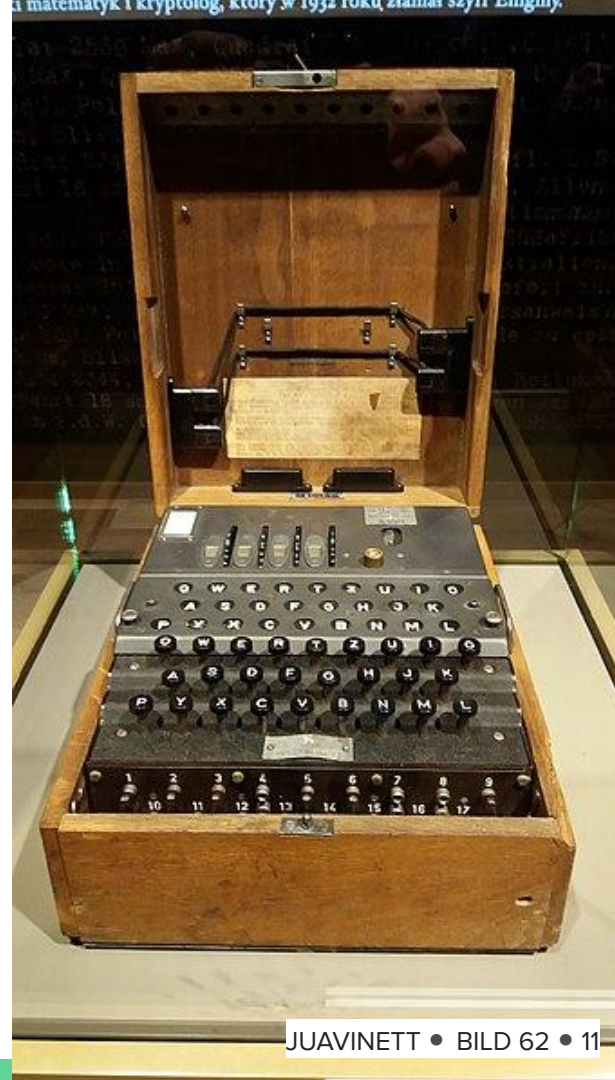
Dr. Andrew Esty

Time of Death: 03/16 11:53

Cause of Death: Rounding Errors

Historical sidenote

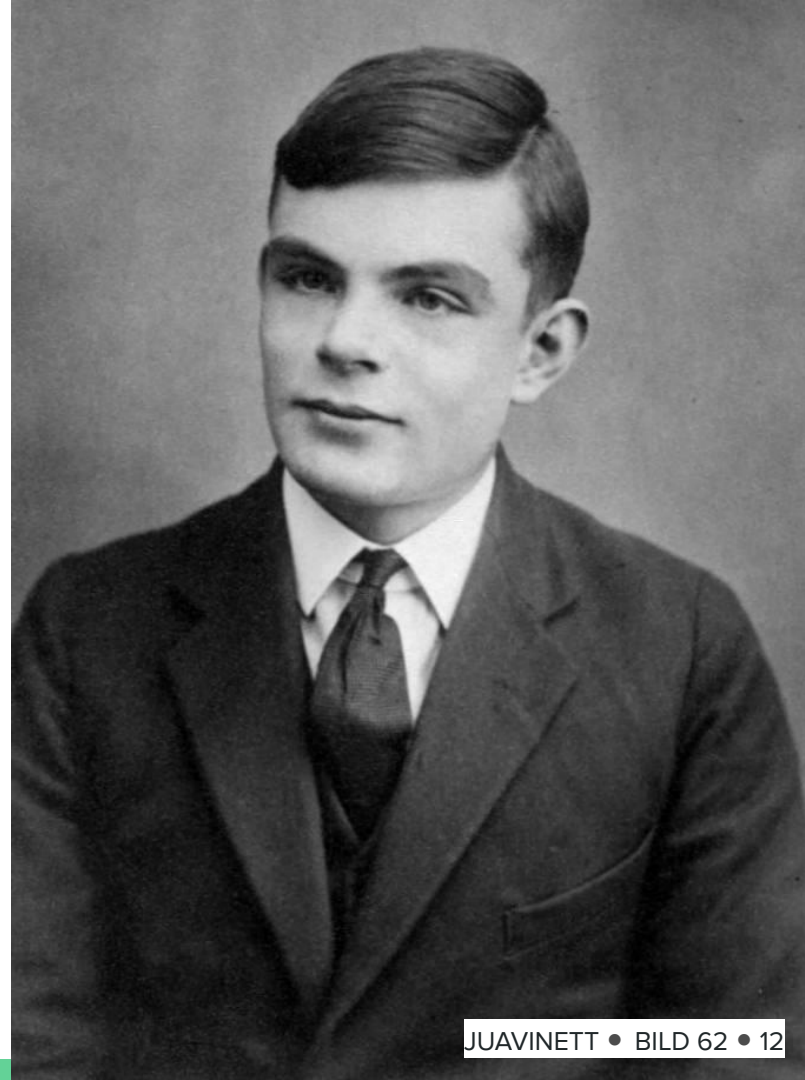
- A **cipher** is a procedure for **encoding** and **decoding** messages or data.
- By manipulating strings, it's quite straightforward to create an encoder using a few lines of code.
- During WWII, Germany used a **variable encoder** which changes its encoding strategy each time it runs.



Historical sidenote

- A team led by **Alan Turing** built & programmed machines that could crack the ENIGMA code, ultimately shortening the war & saving many lives.
- Alan Turing went on to make many contributions to computer science until he was prosecuted by the British Government for homosexual activity, and ultimately committed suicide at 41 years of age.

Note: The Imitation Game is an Oscar-worthy depiction of his life and work (but takes some dramatic liberties...)



Let's dig in to notebook 03!

1. Use the [magic link](#) to sync up your DataHub with our Materials folder and open 03.
2. Open up the Expressions quiz on Canvas

Topics from today

- Assigning variables & rules for variable names
- Types of objects we met: integers, floats, strings
- How to **concatenate** and **slice** strings
- Functions we learned: `type`, `isinstance`, `float`, `int`, `str`

Resources

[A List of Good Python YouTube Channels](#)

[CodeAcademy Python Syntax Cheatsheet](#)

[Software Carpentry: Python Fundamentals](#)

[Software Carpentry: Variables & Assignment](#)

[Software Carpentry: Data Types & Type Conversion](#)

[Error types in Python](#)