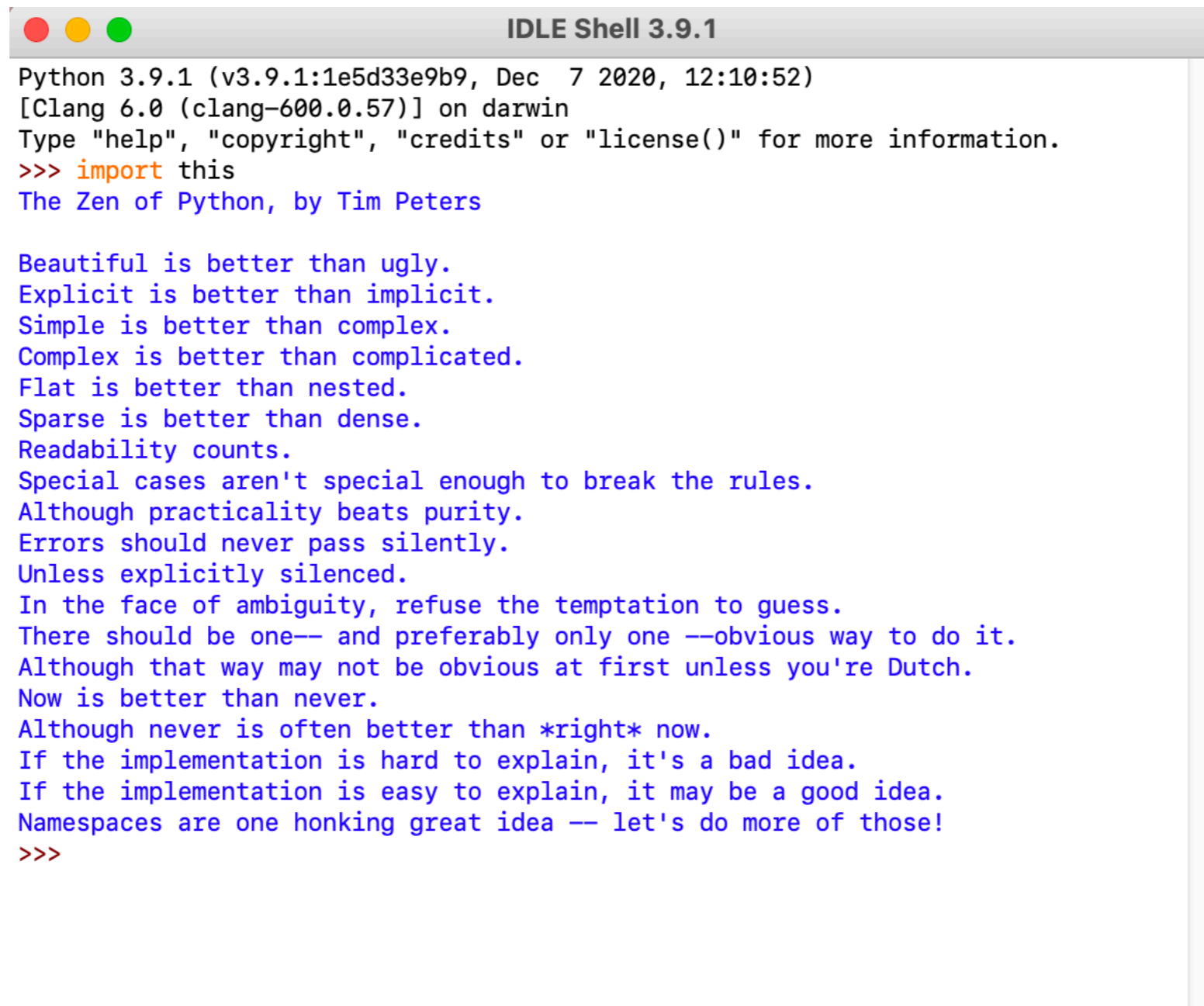


Week 1 Exercises

Thomas Schwarz, SJ

The Zen of Python

- Use `import this`



```
Python 3.9.1 (v3.9.1:1e5d33e9b9, Dec 7 2020, 12:10:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

Refresher: Operations

- Write a script with two variables, x and y
 - Set these variables to 6 and 7, resp.
 - Print out: sum, difference, product, division, floor division, power, remainder, negatives, absolute values (`abs`)
- Note: There are more operations, the binary operations
 - $x | y$ (bit-wise or), $x ^ y$ (bit-wise exclusive or), $x \& y$ (bit-wise

Refresher: Booleans

- Write a script that prints out Boolean comparisons in terms of x and y
 - e.g.: `print(x+y < 10)`

Refresher: Strings

- Strings are created by using simple quotation marks or double quotation marks
- To include line breaks:
 - Use triple quotes or the escape sequence `\n`
- Translate the following code to one using `\n`:

```
>>> astring = """one line
two lines
three lines
"""
>>> print(astring)
one line
two lines
three lines
```

Strings

- Python has a lot of methods working on strings
 - Length — len
 - Concatenation and repetition with + and *
 - You can change to lower or to upper case:

-

```
>>> astring.upper()
'ONE LINE\nTWO LINES\nTHREE LINES\n'
>>> print(astring.upper())
ONE LINE
TWO LINES
THREE LINES

>>> |
```

Strings

- There are a lot of useful string methods
 - Finding / matching

- ```
>>> astring.find('Two')
-1
>>> astring.find('two')
9
```

```
>>> astring.startswith('one')
True
```

```
>>> 'two' in astring
True
```

```
>>> astring.endswith('lines')
False
```

- Length: `len`
- Replace

# Python Survival Rules

- Python Error Messages
  - When you program seriously, you make serious mistakes, seriously
  - So, you better start understanding error messages



# Python Error Messages

- Typically, you call functions within functions within functions ...
  - Traceback lists all function calls
- Because we are not using functions yet, this is simple:
  - Traceback is a single line

# Python Error Messages

- Traceback lists the offending line number
  - BUT: this is when Python notices the error
- Error messages are difficult to read

```
Enter a number for me to convert 3
Traceback (most recent call last):
 File "/Users/thomasschwarz/Documents/My website/Classes/PDS2021/Week 1 Exercises/example1.py", line 2, in <module>
 print(3/(3-int(user_input)))
ZeroDivisionError: division by zero
>>>
```

# Python Error Messages

```
Enter a number for me to convert 3
Traceback (most recent call last):
 File "/Users/thomasschwarz/Documents/My website/Classes/PDS2021/Week 1 Exercises/example1.py", line 2, in <module>
 print(3/(3-int(user_input)))
ZeroDivisionError: division by zero
>>>
```

- The error has a name: ZeroDivisionError
  - Can google that with key-word Python
- It also has a description, but it is a bit laconic
  - Here: user input was 3, so we were dividing by zero

# Sources for Help

- Internet searches for the problem

NEVER HAVE I FELT SO  
CLOSE TO ANOTHER SOUL  
AND YET SO HELPLESSLY ALONE  
AS WHEN I GOOGLE AN ERROR  
AND THERE'S ONE RESULT  
A THREAD BY SOMEONE  
WITH THE SAME PROBLEM  
AND NO ANSWER  
LAST POSTED TO IN 2003

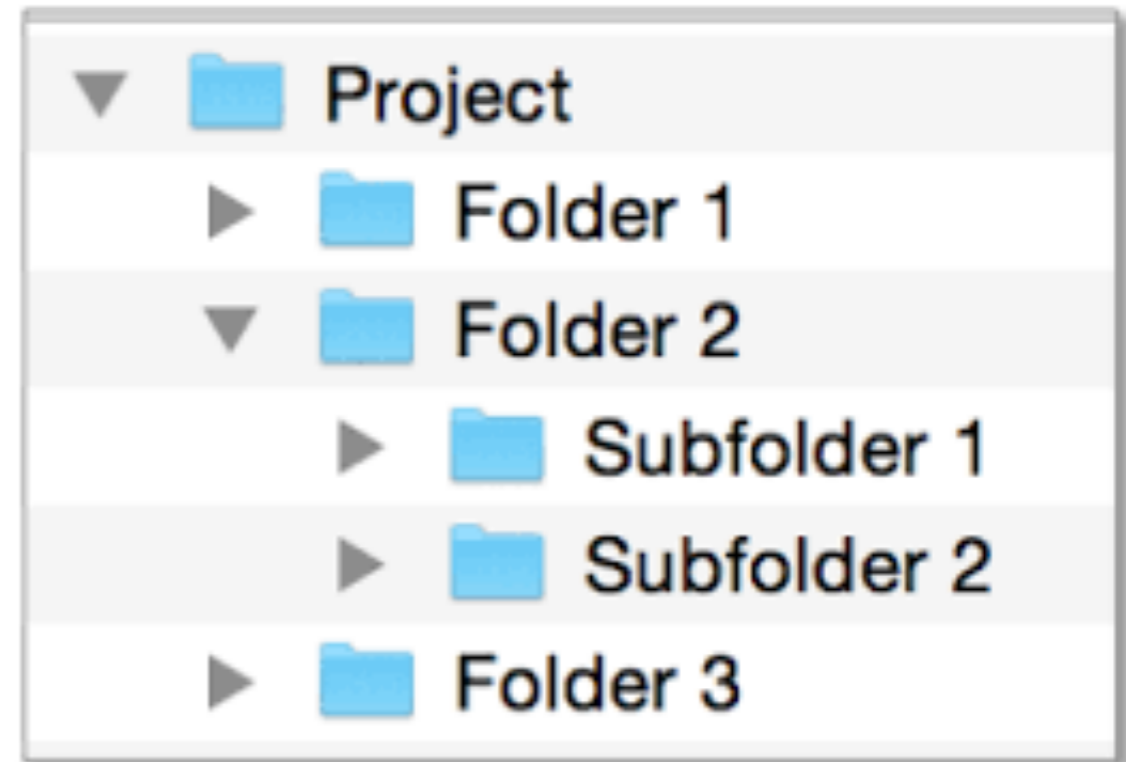


# Sources for Help

- Stackoverflow
  - Answer quality ranges from acceptable to superb
  - Asking questions:
    - First, check that there is no answer for your problem
    - Second, check the guidelines
      - <https://stackoverflow.com/help/how-to-ask>

# Environment Setup

- Sooner or later, you will need to include files
- Windows:
  - Paths start with a drive name
  - Backslashes separate folders and file names



- `c:\\Project\\Folder2\\Subfolder1\\my_file`
- Sometimes, you need to escape

# Environment Setup

- On Mac and Linux, use a forward slash
  - There is no drive letter

`/Users/thomasschwarz/Documents/Marquette Letterhead/Letterhead/ts.pdf`

# Environment Setup

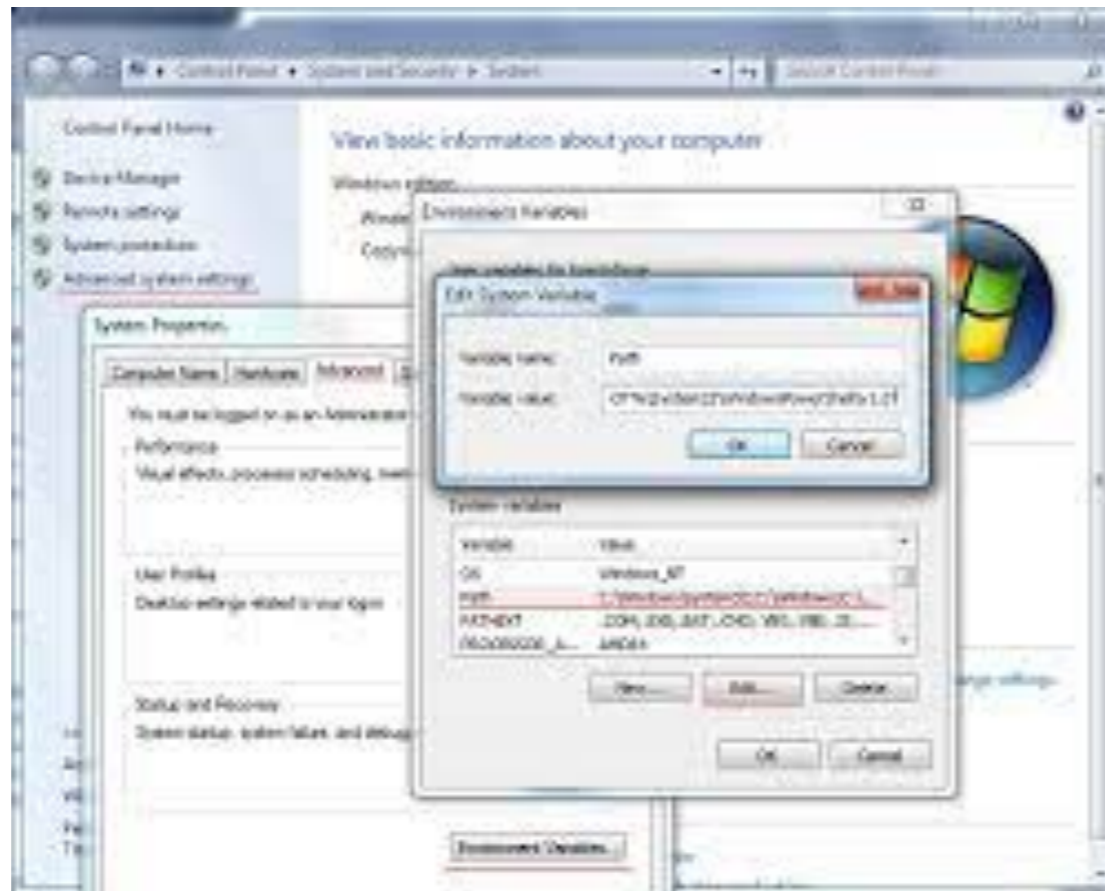
- Whenever you run a program, this program runs in an environment
  - Characterized by environmental variables
    - To see them: Use "set" in a command prompt on windows
    - On Mac/Unix/Linux: use "env"

```
[thomasschwarz@Peter-Canisius Letterhead % env
TMPDIR=/var/folders/18/jn59j8pd04xbj0jp_b6tv3gw0000gq/T/
__CFBundleIdentifier=com.apple.Terminal
XPC_FLAGS=0x0
TERM=xterm-256color
DISPLAY=/private/tmp/com.apple.launchd.BiiX72dWpg/org.macosforge.xquartz:0
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.pxNjaXhUju/Listeners
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=440
TERM_SESSION_ID=5F2FC48B-50DD-41A7-B538-278922964B33
SHELL=/bin/zsh
HOME=/Users/thomasschwarz
LOGNAME=thomasschwarz
USER=thomasschwarz
PATH=/Library/Frameworks/Python.framework/Versions/3.9/bin:/Library/Frameworks/Python.framework/Versions/3.8/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/texlive/2018/bin/x86_64-darwin:/Library/TeX/texbin:/usr/local/share/dot
```



# Environment Setup

- One of the more important variables is the PATH
  - This is the set of locations that the OS looks through when it sees a command
  - On Windows, path variables are separated by semi-colons



# Environment Setup

- On Mac / Linux, its colons

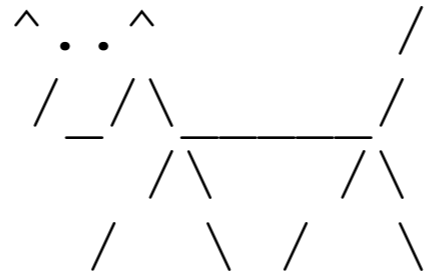
```
echo $PATH
/Library/Frameworks/Python.framework/Versions/3.9/bin:/Library/
Frameworks/Python.framework/Versions/3.8/bin:/usr/local/bin:/usr/bin:/
bin:/usr/sbin:/sbin:/usr/local/texlive/2018/bin/x86_64-darwin:/
Library/TeX/texbin:/usr/local/share/dotnet:/opt/X11/bin:~/dotnet/
tools:/Library/Apple/usr/bin:/Library/Frameworks/Mono.framework/
Versions/Current/Commands:/Applications/Wireshark.app/Contents/MacOS
```

# Exercises

- Hero's algorithm calculates the square-root of a number  $S$ .
- First, estimate the square root  $x_0$  (or set it to one)
- Second, improve on your guess by calculating
  - $x_{\text{next}} = \frac{1}{2} \left( x_{\text{old}} + \frac{S}{x_{\text{old}}} \right)$
- Repeat (until approximation is good)

# Exercises

- ASCII art: Use escape characters to build the following animal

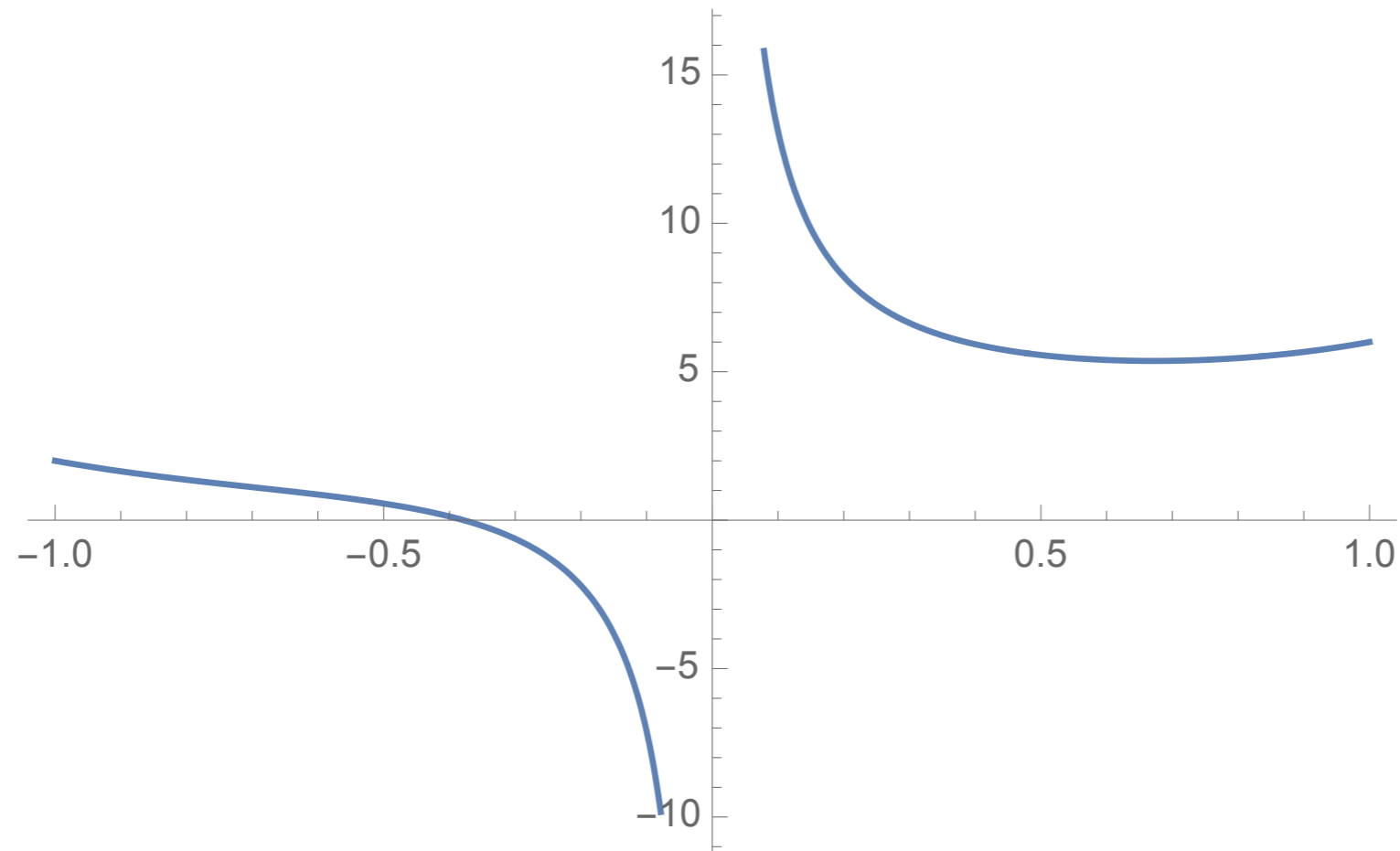


# Exercises

- Create a Python script that asks a user for a distance in furlongs and prints out the distance in meters
  - (one furlong is 201.1684 meters)
  - You can use the Python `round` function to change the floating point number to

# Exercises

- Assume that we want to solve an expression such as  $x^4 + x + 3 + 1/x = 0$ .



# Exercises

- Assume that we want to solve an expression such as  $x^4 + x + 3 + 1/x = 0$ .
  - For  $x = -0.1$  the value is negative
  - For  $x = -0.5$ , the value is positive
  - We "bisection" the interval by taking middle
    - Since with  $b = -0.3$ ,  
 $b^4 + b + 3 + 1/b \approx -0.625233$ ,
      - The zero lies between  $a = -0.5$  and  $b = -0.3$

# Exercises

- We bisect the interval again:
  - $x = -0.4$
  - Then  $x^4 + x + 3 + 1/x \approx 0.1256$
- We know the zero is between  $a = -0.4$  and  $b = -0.3$
- We then divide the interval again:
  - $x = -0.35$ .
  - Then  $x^4 + x + 3 + 1/x \approx -0.192137$
- Thus, the zero is between  $a = -0.4$  and  $b = -0.35$



# Exercises

- And now we continue

# Exercises

- Calculate the following sums and products

- $$\sum_{i=1}^{1000} i^4$$

- $$\sum_{i=0}^{1000} \frac{1}{1+i^2}$$

- $$\sum_{i=1}^n \frac{1}{i^2}$$

# Exercises

- Calculate the following sums and products

- $$\prod_{i=1}^{1000} \frac{2i+1}{2i}$$

# Exercises

- Find the two three digit numbers such that  $\frac{a}{b} \approx \pi$ 
  - Hint:  $\pi$  is `math.pi`, but you need to `import math`
  - Hint: Generate all three digit numbers a and b
    - Calculate the absolute difference of a/b and `math.pi`
    - If this is better than the previous best value, remember the absolute difference and the best values for a and b.