Day - 13

**Debugging**

**How to Find and Fix Errors in your Code**

Tips for debugging code

**13.1 Debugging**

Debugging: Process of removing bugs from the code.

**Techniques and tips to find bugs and get rid of them**

* Use comment, un-comment: **ctrl + /**

**13.2 Describe the Problem**

Understand the problem clearly, how should it work. Possibilities why not working.

|  |  |
| --- | --- |
| #*---------------------  DEBUGGING -----------------------*  #*follwing has bugs: Debug it*  #*def my\_function():*  #*for i in range(1, 20):*  #*if i == 20:*  #*print("You got it")*  #*my\_function()* | #*for loop range, upper-bound need set to 21*  **def** **my\_function**():  **for** i **in** **range**(1, 21):  **if** i **==** 20:  **print**("You got it")  **my\_function**()  #*python describe.py* |

**13.3 Reproduce the Bug**

Some bugs appear and disappear randomly. It is difficult to find this kind of bug. So reproduce the bug to really understand it.

|  |  |
| --- | --- |
| #*----------------------  DEBUGGING ------------------------*  #*Debug following by reproducing the bug*  #*from random import randint*  #*dice\_imgs = ["A", "B", "C", "D", "E", "F"]*  #*dice\_num = randint(1, 6)*  #*print(dice\_imgs[dice\_num])*  #*Using for loop to find the bug. Index out range will appear eventually*          #*Traceback (most recent call last):*          #*File "reproduce.py", line 16, in <module>*          #*print(dice\_imgs[dice\_num])*          #*IndexError: list index out of range* | #*Problems are : A doesn't appear, Index error appear.*  #*Cahnge the rand int : randint(1, 6) to randint(0, 5)*  """  for i in range (0, 19):    from random import randint    dice\_imgs = ["A", "B", "C", "D", "E", "F"]    dice\_num = randint(0, 5)    print(dice\_imgs[dice\_num])      """    **from** random **import** randint  dice\_imgs = ["A", "B", "C", "D", "E", "F"]  dice\_num = **randint**(0, 5)  **print**(dice\_imgs[dice\_num]) |

**13.4 Think as a COMPUTER/COMPILER/Interpreter**

Imagine that you are a computer, and think like computer to evaluate the code step by step.

#*----------------------  DEBUGGING ------------------------*

#*let's say input year is 1994, it deosn't belong to any of the condition.*

#*if you put it in a if statement, 1994>1980 ? => True, 1994 >1994 ? => False, True and False => Flase*

#*in following code 1994 doesnt print any thing*

  #*the reason is : 1994 is excluded accidentaly in the conditions.*

  #*if or elif both conditions does not includes 1994.*

  #*Solution is include 1994 in any one of those two conditions. Eg: <= 1994*

"""

year = int(input("What's your year of birth?"))

if year > 1980 and year < 1994:

  print("You are a millenial.")

elif year > 1994:

  print("You are a Gen Z.")

"""

year = **int**(input("What's your year of birth?"))

**if** year **>** 1980 **and** year **<=** 1994:

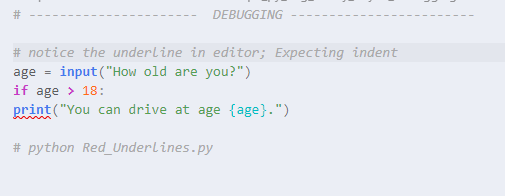
**print**("You are a millenial.")

**elif** year **>** 1994:

**print**("You are a Gen Z.")

#*python act\_as\_computer.py*

**13.5 Fixing Errors on Experience and Red Underlines in Editor**



* Notice the red underline. It noticing that the ***print()*** needs an indent.
* Also notice the ***f-string*** where "***f***" is missing. This bug is hard to find and needs experience to fix it.

**print**("You can drive at age {age}.")

* Google the error-message, use stack overflow to tack down the bug.
* Some bugs are HARD to find and needs experience to find it. Eg: missing "**f**" in f-string is really hard to find.
* How to get Experience:
* Code more and more
* Help other programmer. Eg: become a StackOverflow helper

#*----------------------  DEBUGGING ------------------------*

#*1. Notice the underline in editor; Expecting indent*

#*2. Also type-coversion is needed. Somtimes googling error messges works.*

        #*Use "stackoverflow"*

#*3. Fixing above will works perfectly, But also there is a bug. Notice "f" missing in f-string*

"""

age = input("How old are you?")

if age > 18:

print("You can drive at age {age}.")

"""

age = **int**(input("How old are you?"))

**if** age **>** 18:

**print**(f"You can drive at age {age}.")

#*python Red\_Underlines.py*

**13.6 Narrow-down bugs with** print() **Statement**

* **print()** is Your Friend
* ***print()*** variables from time to time
* Use ***print()*** anywhere where you think the bug may occur.

#*----------------------  DEBUGGING ------------------------*

#*Print is Your Friend*

#*Print every result to Narrow down the bug*

"""

pages = 0

word\_per\_page = 0

pages = int(input("Number of pages: "))

word\_per\_page == int(input("Number of words per page: "))

total\_words = pages \* word\_per\_page

print(total\_words)

"""

#*Notice the "word\_per\_page"  is using a logical equal sign "==". This is the bug*

#*Evaluated as True or False*

pages = 0

word\_per\_page = 0

pages = **int**(input("Number of pages: "))

#*print(word\_per\_page == int(input("Number of words per page: ")))*

word\_per\_page = **int**(input("Number of words per page: "))

total\_words = pages \* word\_per\_page

**print**(total\_words)

#*python use\_of\_print.py*

**13.7 Using a Debugger**

* Debugger like ***thonny*** or "***pythontutor.com***" can be very useful.
* These debugger shows step by step evaluation of code.
* We also can set a ***break-point*** and analyze the steps toward the ***break-point*** to catch the bug.

<http://pythontutor.com/visualize.html#mode=display>

For example to fix following codes bug we need an "indentation"

|  |  |
| --- | --- |
|  | #*------------------  DEBUGGING -------------------*  #*Use a Debugger*  """  def mutate(a\_list):    b\_list = []    for item in a\_list:      new\_item = item \* 2    b\_list.append(new\_item) #indentation needed    print(b\_list)  mutate([1,2,3,5,8,13])  """  **def** **mutate**(a\_list):    b\_list = []  **for** item **in** a\_list:      new\_item = item \* 2      b\_list**.append**(new\_item) #*indentation needed*  **print**(b\_list)  **mutate**([1,2,3,5,8,13])  #*python use\_debugger.py* |

**13.8 Some more TIPS**

1. Take a Break: if you starting at the code for a long time, then your brain won't work. Have some down time, and come back to it, everything gets so obvious.
2. Ask a Friend!: With fresh eyes, your friends can help you.
3. Run Often

* Pile of code generate pile of bugs. Keep your code clean always (don’t wait). Don't wait until you write loads of codes. Run and confirm.
* Don’t leave small bugs: leave it all to the end and end up with pile of bugs.. Neglecting a bug may cause more bugs (which is nightmare).
* Run the code often during development

1. Ask/ Search StackOverflow:

* Bug or issue should be unique. if not, just search. Vital tool!
* You only wanna ask a question when you've pretty sure that you've exhausted all other avenues of debugging and you've searched all of Stack Overflow.
* Exercise 13.1: Debugging Odd or Even

#*----------------------  DEBUGGING ------------------------*

#*Notice the conditinal evaluation*

"""

number = int(input("Which number do you want to check?"))

if number % 2 == 0:

  print("This is an even number.")

else:

  print("This is an odd number.")

"""

number = **int**(input("Which number do you want to check?"))

**if** number % 2 **==** 0:

**print**("This is an even number.")

**else**:

**print**("This is an odd number.")

#*python Debugging\_Odd\_Even.py*

* Exercise 13.2: Debugging Leap Year

#*----------------------  DEBUGGING ------------------------*

#*Notice the input. It needs type conversion for conditional comaprison*

"""

year = input("Which year do you want to check?")

if year % 4 == 0:

  if year % 100 == 0:

    if year % 400 == 0:

      print("Leap year.")

    else:

      print("Not leap year.")

  else:

    print("Leap year.")

else:

  print("Not leap year.")

"""

year = **int**(input("Which year do you want to check?"))

**if** year % 4 **==** 0:

**if** year % 100 **==** 0:

**if** year % 400 **==** 0:

**print**("Leap year.")

**else**:

**print**("Not leap year.")

**else**:

**print**("Leap year.")

**else**:

**print**("Not leap year.")

#*python Debugging\_Lap\_year.py*

* Exercise 13.3: Debugging FizzBuzz

#*----------------------  DEBUGGING ------------------------*

#*Notice "*or*" is used instead of "and" in first condition*

#*also notice "*elif*" not used. So number is printed along with "fizz-buzz"*

"""

for number in range(1, 101):

  if number % 3 == 0 or number % 5 == 0 :

    print("FizzBuzz")

  if number % 3 == 0:

    print("Fizz")

  if number % 5 == 0:

    print("Buzz")

  else:

    print([number])

"""

**for** number **in** **range**(1, 101):

**if** (number % 3 **==** 0) **and** (number % 5 **==** 0) :

**print**("FizzBuzz")

**elif** number % 3 **==** 0:

**print**("Fizz")

**elif** number % 5 **==** 0:

**print**("Buzz")

**else**:

**print**(number)

#*Modify the code to fix the program.*

#*No shortcuts - don't copy-paste to replace the code entirely with a working solution.*

#*python Debugging\_FizzBuzz.py*