

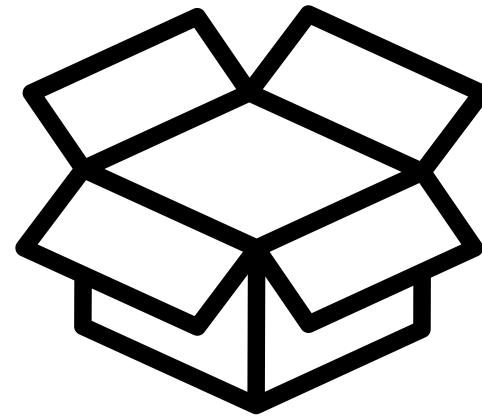
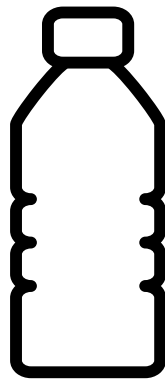
COMP10001 - Sem 2 2024 - Week 3

Foundations of Computing



Daksh Agrawal

Data Types



"135"


"hello" + "world" hello

bool(123)=True
bool(-32)=True
bool(0)=False

Identifying Data Types

Type	Example	What does it store?	What can we do with it (functions, operations...)?	How do we convert to it?
string	"Hello"	text	print concatenation slicing, indexing	str(~)
integer	123	integers (no decimals)	+ - x ÷ >> <<	int(~)
float	3.1415	real numbers	+ - x ÷ >> << ><	float(~)
boolean	True False	Truth values	and or not	bool(~)

Library Database

<div>Name</div> <div>text strings</div>	<div>Late Fees Owed</div> <div>float 13.45 14.5555555 int (pennies)</div>	<div>Whether they are a student</div> <div>boolean "student" "staff" "external"] string</div>
<div>Number of Books loaned out</div> <div>integer</div>	<div>User ID</div> <div>integer strings</div>	<div>Borrow Date and Time</div> <div>strings integers</div> <div></div>

Evaluate by Hand

`str(3 + 4) + "cakes"`

`str(7)`

`"7"`

`"7cakes"`

`float("357" + "." + "23")`

`"357.23"`

`357.23` float

`int(5 / 2)`

`→ 2.5`

`2`

`bool("anything")`

`True`

Evaluate by Hand, given $a = 1$, $b = 2$, $c = 2.0$

a / a

1 / 1
1.0 (float)

$b + b$

2 + 2
4 (int)

$b + c$

2 + 2.0
4.0 (float)

a / b

1 / 2
0.5 (float)

$a // b$

floor division
1 // 2
0 (int)
quotient

$a \% b$

modulo
remainder
1 (int)

$a + b / c$

1 + 2 / 2.0
1 + 1.0
2.0
float

$(a + b) / c$

(1 + 2) / 2.0
3 / 2.0
1.5
float

Evaluate by Hand

123 + 123

246 (int)

"123" + "123"

"123123"
string

"123" + 123

~~Error~~
"123" + str(123)

3 * 4

12 (int)

"3" * 4

"3333"
string

"3" * "4"

~~Error~~

IDLE Shell 3.12.2

Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more
information.

```
>>> 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1  
0.9999999999999999
```

```
>>> 1000000000000 * 0.000000000001  
0.9999999999999999
```

```
>>> |
```

$$0.1 = 2^x$$

Ln: 7 Col: 0

rounding error

(ALL LANGUAGES)

a	b	a and b	a or b	not a
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

Truth Tables

Logical Methods

Time for the Truth

NO

True **or** False

True (bool)

True **and** False

False (bool)

False **and** not False **or** True

False and True, or True

False or True

True

False **and** (not False **or** True)

False and (True or True)

False and True

False

```
if                      :  
    → print ("True")  
else:  
    → print ("False")
```

If you can dream—and not make dreams your master;

```
if 10 > var >= 5:  
    print("Hello")
```

5,6,7,8,9 ✓
1,2,3,4,10,11 X

```
if var[0] == "A" and var[-1] == "e":
```

"Are" ✓
"Acre"

"hello" X
"Arrow" X
"House" X

```
if var in ("VIC", "NSW", "ACT"):
```

"VIC"
"NSW" ✓
"ACT"

"SA" X
"TAS" X

```
if var:
```

True ✓
1 ✓
"hello" ✓
232 ✓

False X
0 X
None X

Fix the Code

eggs == 3 eggs = 3

if eggs = 5:

 print("spam")

else:

 print("not spam")

eggs = 3 ← assignment

if eggs == 5: ← comparison
 print("spam")

else:
 print("not spam")

== comparison

= equality assignment

a = "house" ← statement

a == "house" ← True
 ↑ False
 question

Fix the Code

```
letter = input("Enter a letter: ")
```

```
if letter == 'a' or 'e' or 'i' or 'o' or 'u':  
    print("vowel")
```

```
else:
```

```
    print("consonant")
```

if letter == "a", "e" ✗

if letter == "a" or letter == "e" or letter == "i", ... :

if letter in ("a", "e", "i", "o", "u"):

$a[0:7:2] = \text{"uie b"} = a[: :2]$
 $a[6:-1:-1] = \text{"blemium"} = a[: :-1]$

A diagram illustrating the indexing of the string "unimelb". The string is written in blue. Above each character is its positive index (0 to 6) in green. Below each character is its negative index (-7 to -1) in green. A red bracket above the string spans from index 0 to index 6, with an arrow pointing left towards the first bracketed slice operation in the text above.

0	1	2	3	4	5	6
u	n	i	m	e	l	b
-7	-6	-5	-4	-3	-2	-1

Slicing

$a[0:3] = \text{"uni"} = a[:3]$

$a[3:7] = \text{"melb"} = a[3:]$

$a =$
Indexing
 $a[0] = \text{"u"}$
 $a[3] = \text{"m"}$
 $a[-1] = \text{"b"}$
 $a[-5] = \text{"i"}$

Evaluate by Hand, given `s = "Python"`
_{0 1 2 3 4 5}

^{-6 -5 -4 -3 -2 -1}
Python
_{0 1 2 3 4 5}

`s[1]`

"y"

`s[-1]`

"n"

`s[1:3] + s[3:5]`

"ytho"

`s[10]`

Error

`s[10:]`

""

`s[::-2]`

"Pto"

`s[1:3] + s[3:5]`

"yt" + "ho"
"ytho"

`s[-4:-2]`

"th"

`s[::-1]`

"nohtyP"

Paper Programming



Write a program which asks the user for their age and calculates the year in which they were born. There will be two possibilities since you haven't asked for their birthdate, so print both. For example, your program should work similar to this when the user inputs 18:

```
Enter your age: 18
```

```
You were born in either 2006 or 2005
```

Write a program which asks the user for a temperature in degrees Fahrenheit and prints the corresponding value in Celsius. The conversion formula is below:

$$C = \frac{F - 32}{1.8}$$

This is an example of how the program could work when the user inputs 90:

```
Enter the temperature in Fahrenheit: 90
90.0 Fahrenheit converts to 32.222222222222222 Celsius
```

Write a program which asks the user for a word, and prints a shortened version of their word consisting of its first three letters and then every second letter in the rest of the word. For example, when the user inputs the word Honorificabilitudinitatibus, the program might work like this:

```
Enter a word: Honorificabilitudinitatibus
Honrfcbltdnttbs
```

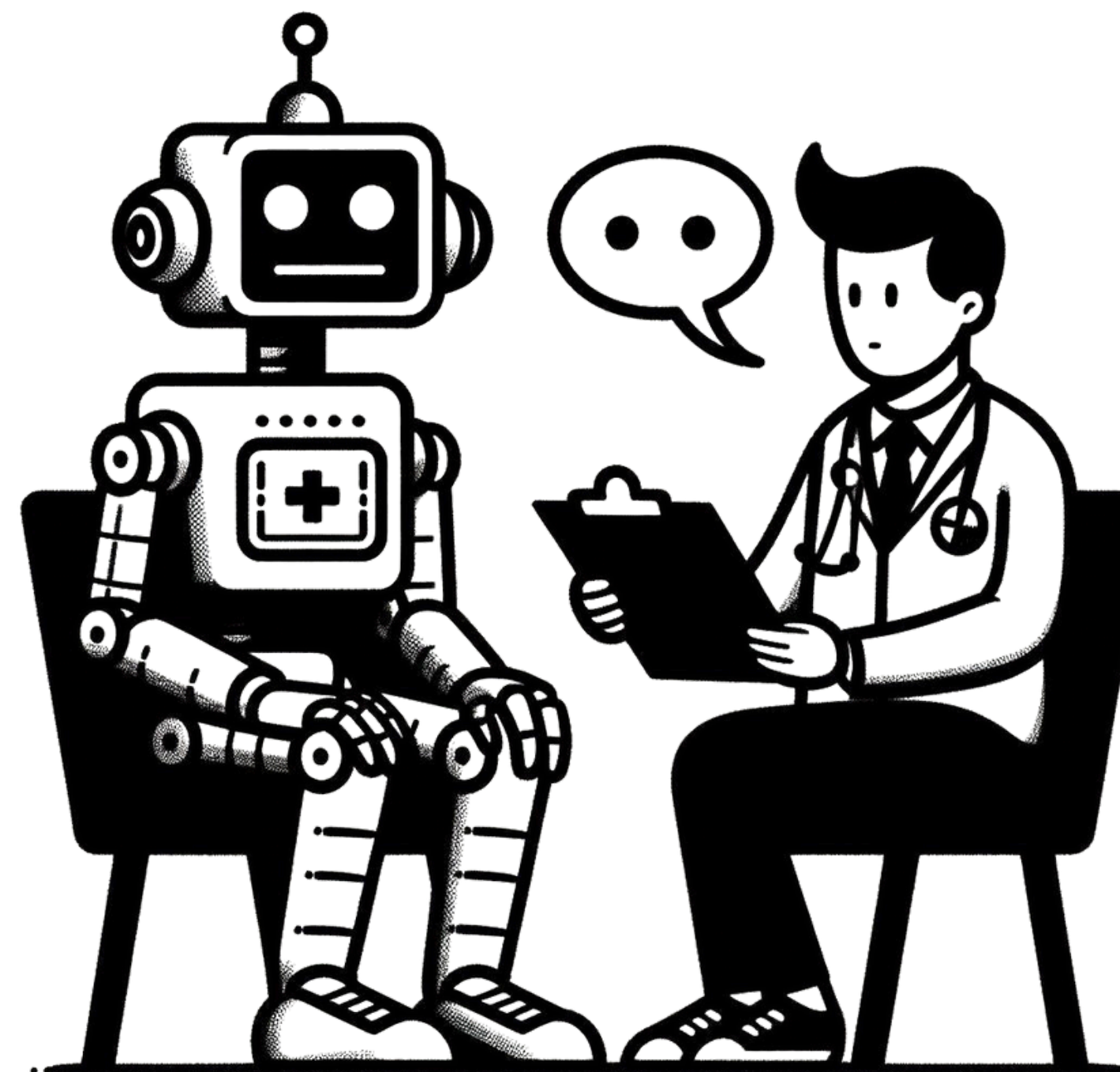
Heat Waves

In this question you'll be writing a program that asks the user about the weather and prints snappy responses that depend on what the user has input. Feel free to be creative in the responses and try to come up with at least four (including one for an `else` branch)! Some examples of how your program could behave:

What's the weather like? `rainy`
Raining cats and dogs!

What's the weather like? `typical Melbourne`
Four seasons in one day!

Meet ELIZA



Can you create a basic ELIZA?

Grok Worksheets 0-2 due soon

Good time to start Worksheets 3-5

