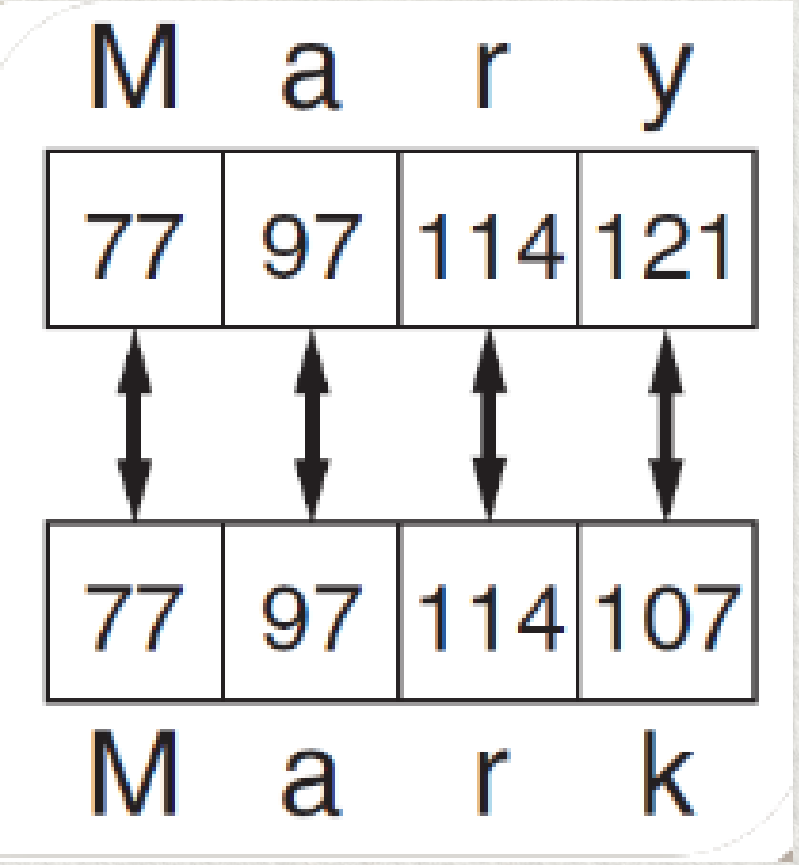
Lecture 9
Week 5

String comparison

Strings

Using Strings in selection statements



- Strings can be compared using >, <, >=, and <=
- Compared character by character based on the ASCII values for each character
- If shorter word is has all the letters of a longer word, the longer word is greater than shorter word e.g. "Sur" is less than "Sure"

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

SAMPLE CODE

```
word1 = input("What is the first word? ")
word2 = input("What is the second word? ")
if word1 == word2:
    print("They are the same")
elif word1 > word2:
    print(f"{word2} comes before {word1} in the alphabet.")
elif word1 < word2:
    print(f"{word1} comes before {word2} in the alphabet.")
```

```
What is the first word? dog
What is the second word? cat
cat comes before dog in the alphabet.
```

SAMPLE CODE

```
word1 = input("What is the first word? ")
word2 = input("What is the second word? ")
if word1 == word2:
    print("They are the same")
else:
    if word1 > word2:
        print(f"{word2} comes before {word1} in the alphabet.")
    if word1 < word2:
        print(f"{word1} comes before {word2} in the alphabet.")
```

NESTED IF STATEMENT

```
What is the first word? dog
What is the second word? cat
cat comes before dog in the alphabet.
```

Dealing with case

Alternatively use .upper() to uppercase both words or .capitalize() to ensure the first letter is uppercased and all other letters are lowered.

Dealing with case

- Let's imagine we want to ignore case so this won't happen

```
What is the first word? cat
What is the second word? Cat
Cat comes before cat in the alphabet.
```

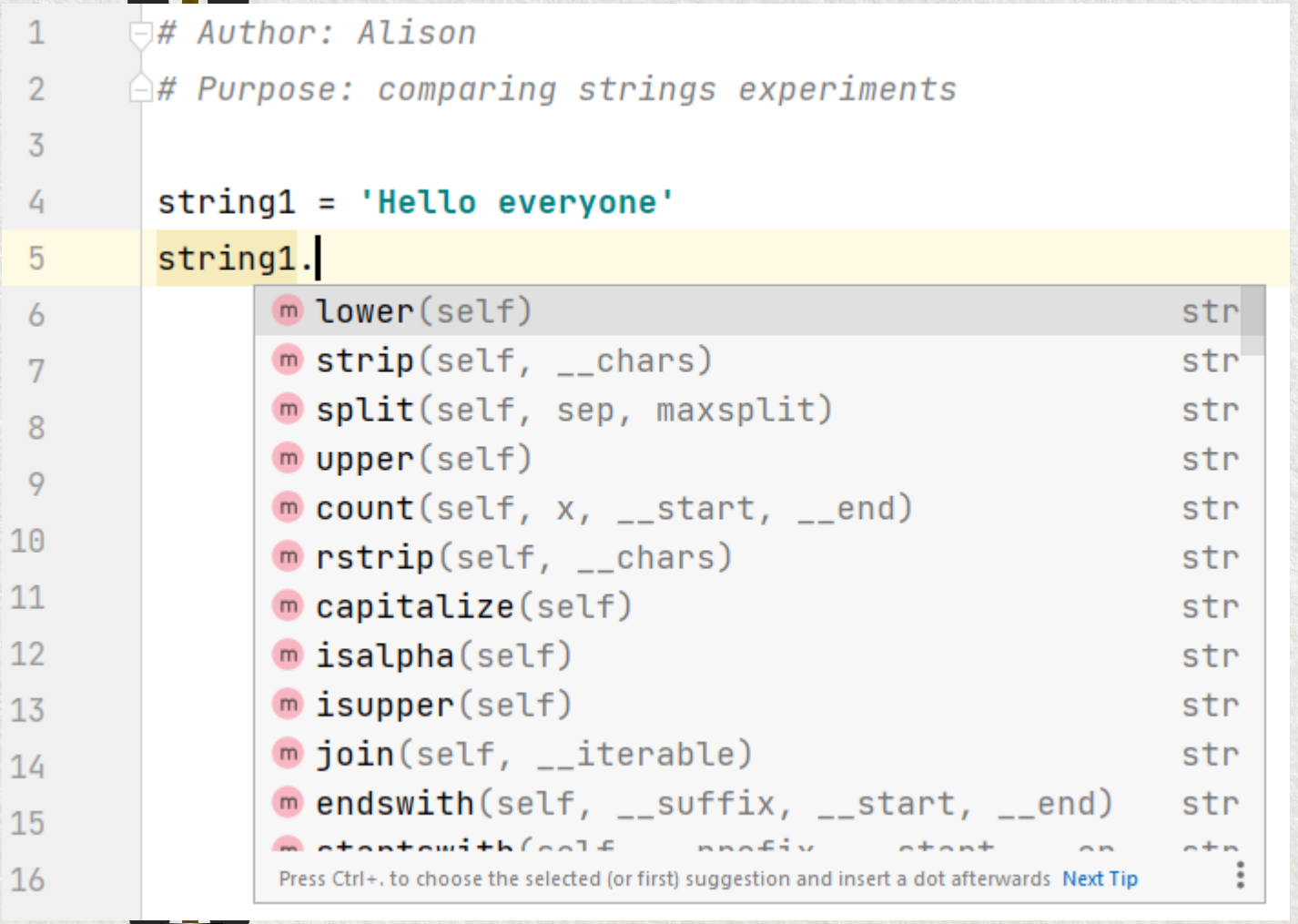
```
word1 = input("What is the first word? ")
word2 = input("What is the second word? ")

# convert them both to the same case
# to compare like with like
word1 = word1.lower()
word2 = word2.lower()

if word1 == word2:
    print("They are the same")
elif word1 > word2:
    print(f"{word2} comes before {word1} in the alphabet.")
elif word1 < word2:
    print(f"{word1} comes before {word2} in the alphabet.")
```

Special string methods

Strings have special built-in hidden abilities.
We can find them using the dot in Pycharm.



Special string methods

`.lower()` gives us a version of the string with all letters in lower case.

It leaves non-alphabetical characters unchanged.

Alternatively use `.upper()` to uppercase both words or `.capitalize()` to ensure the first letter is uppercased and all other letters are lowered.

Boolean string methods

`.endswith()` receives a string in the brackets – it takes a string as a parameter - and must determine if the word ends with that string.

Returns True if the string ends with the given string and False otherwise.

Special string methods

```
word = "Hello"

# true if word starts with "H"
print(word.startswith("H"))
# true if word starts with "Hel"
print(word.startswith("Hel"))

# true if word consists of letters and numbers
print(word.isalnum())

# true if word consists of digits
print(word.isnumeric())
```

```
55 isnumeric True
55 isdecimal True
55 isdigit True

5.5 isnumeric False
5.5 isdecimal False
5.5 isdigit False

½ isnumeric True
½ isdecimal False
½ isdigit False
```

For our purposes these 3 methods are the same so use whichever you prefer.

len()

- len() is a standard python function (no import needed)
- Returns the number of elements in an object
- In a string it returns the number of characters

Get a single character

word = "Fred"
word[0] → "F"
word[1] → "r"
etc.

Company ID

In a company, an ID is 8 characters long and starts with 'S'.

- Write code to read an id from the user.
- Add code to display an error message if the ID is not 8 characters long.
- Add code to display an error message if the ID Is 8 characters long but does not start with 'S'.

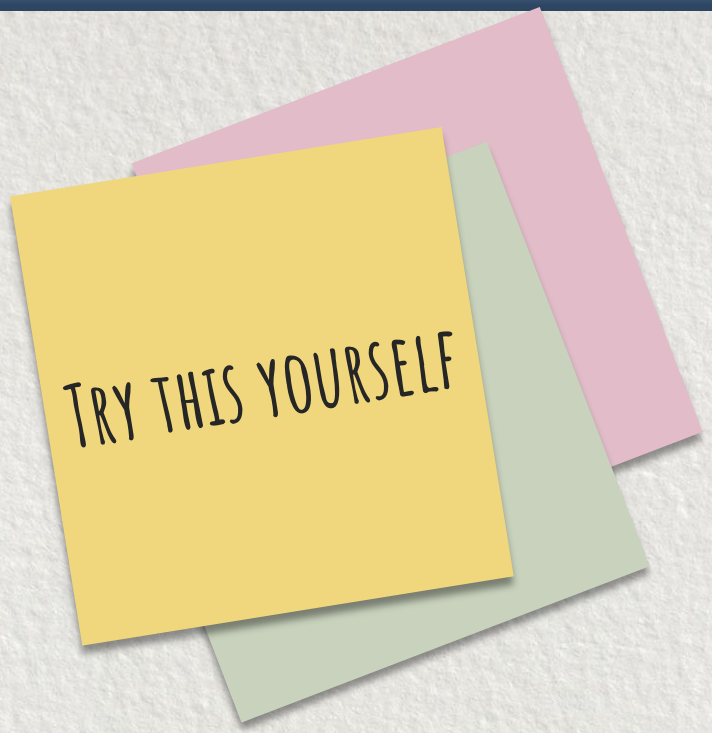
Using % (Modulus) in Boolean expressions

- How would we write code to determine if a number is even i.e. divisible by 2
- How could we update the code to determine if a number is even and it is greater than 10

Wages

Write a program to read a salary from the user.

- If the salary is less than €30,000, the user is not eligible to apply for a loan.
- Otherwise, the user is eligible to apply for a loan if she has been in her job for 3 or more years.



3

2

1

Finished

Starting a Python program

Code order

```
# Author: Who wrote the code
# Purpose: Why are you writing this code? What does it do?

import math
import random

NAMED_CONSTANT = 3
GRAVITY = 9.81

print("The python program is starting")
input_value = input("We are learning about code layout: ")

```