Intro to Computer Science
CS-UH 1001, Spring 2022

Lecture 3 – Input, Characters, Strings

Today's Lecture

- Python Input
- Characters
- Strings
- String Indexing/Slicing
- String Methods

Recap

- Python has 4 simple data types:
  - int
  - float
  - str
  - boolean
- Variables contain 3 pieces of data:
  - Reference
  - Data type
  - Value
- Overloaded operators
  - The + operator does concatenation for strings
  - The * operator does repetition for strings

Recap

- Create variables using an assignment statement
  - number = 10
- Change the value of a variable using a re-assignment statement
  - number = 40
    or
  - number = number * 2

Recap

- The sep="" argument overwrites the , separator of a print() function

- The end="" argument overwrites the line ending of a print() function

Python Input

- Programming often requires input from users to interact with the program

- The input() function is used in Python to read data from the keyboard

  >>> input(message)

- The input() function cycle:
  - Prints the string given in the message argument
  - Waits for input from user (until enter/return)
  - Returns what was typed as a string!!

Python Input

- You should assign/store the input into a variable and use it later

```
>>> name = input("Please enter your name: ")
```

- Even if you enter a number, the input will still be a string!!

- If you want to use the input as a number you have to type cast it:
  - int() or float()

Breakout session I:
Input and output

Simple calculator III (ex_3.1.py)

Modify the previous exercise so that the program asks the user to enter the 2 numbers (each at a time)

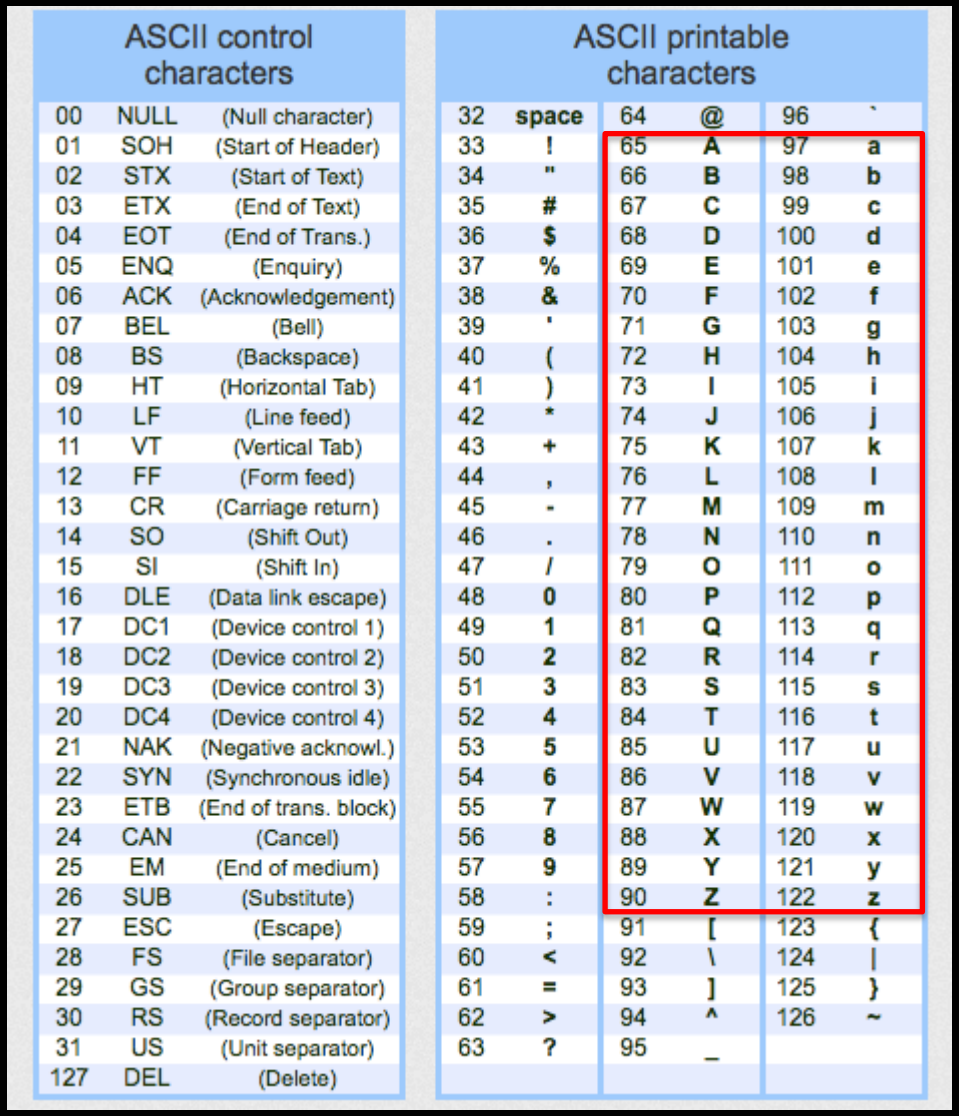
For example, the user enters 2 and 4:

```
>>> Enter the first number: 2
>>> Enter the second number: 4
The sum of 2 and 4 is 6
```

Characters

- Computers do not store characters as characters
  - but rather store their numeric code
- American Standard Code for Information Interchange (ASCII)
  - is a commonly used character coding system
- In ASCII, every letter, digit, and symbol is represented as a number

ASCII Table



ASCII control characters		
00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters					
32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

ASCII-Character Conversion

- Convert a character to its ASCII numerical value:
  >>> number = ord(character)

- Example:
  >>> ord('A')
  65

- Convert an ASCII numerical value to a character:
  character = chr(number)

- Example:
  >>> chr(65)
  A

Recap on Strings

- Strings are one of the data types in Python
- Strings are sequences of characters
- Strings are defined by single or double quotations (‘ ’, “ ”)
- Example:
  >>> name = ‘python’

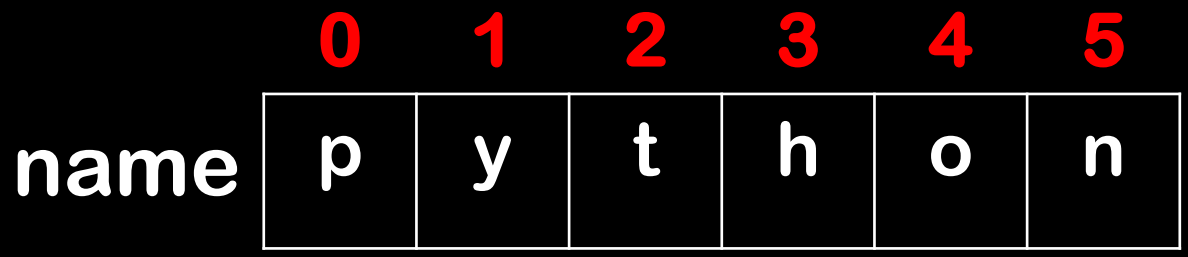
String Indexing

- You can access individual characters by using square brackets after the string:

- >>> character = string[index]

- Arguments:
  - index is an integer representing the index

String Indexing



- What do you think this will give us?

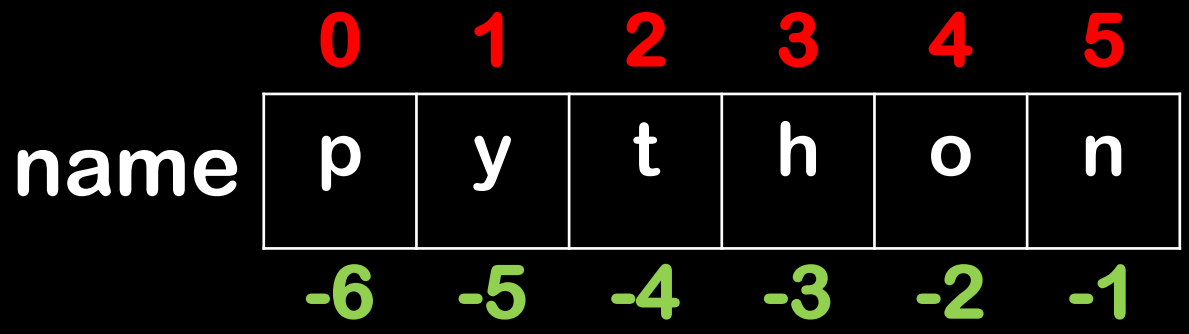
```
>>> name[1]
y
```

- Index is an offset from the beginning of the string
  - So, the offset to the first letter is 0

You know you're a programmer when..

you count 3 apples

String Indexing



- What about the following:

```
>>> name[1.5]    TypeError
>>> name[6]      IndexError
>>> name[-1]
n
```

String Slicing (Substrings)

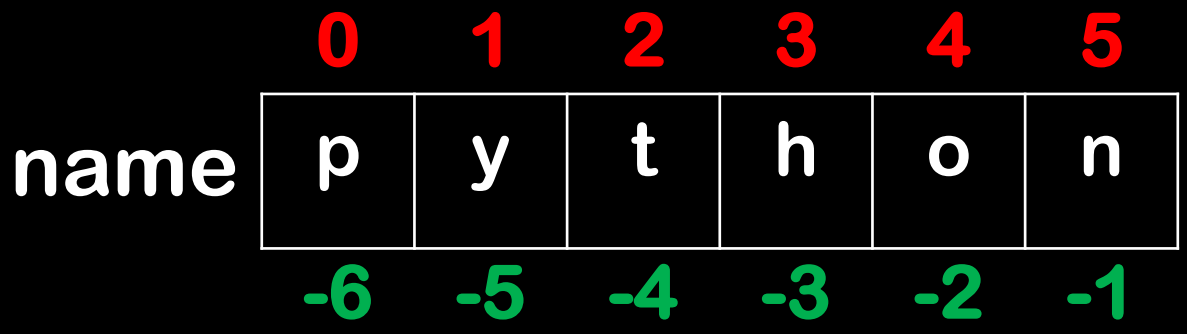
- To get a slice of a string, use the following expression:
  `>>> substring = string[start:end:step]`

- Arguments:
  - start: first character (inclusive) in the slice
  - end: last character (exclusive) in the slice (optional)
  - step: step size (optional)
    - default value +1

- It returns the substring according to the arguments

- start, end and step must be integers and can be positive or negative
- |start| or |end| can be larger than the string length

String Slicing Example




```
>>> name[1]          y
>>> name[-5]         y
>>> name[0:2]        py
>>> name[-6:-1]      pytho
>>> name[-6:]        python
>>> name[-1:-6]      ‘’
>>> name[0:5:2]      pto
>>> name[-1:-7:-1]   nohtyp
>>> name[::-1]       nohtyp
```

Breakout session II:
Strings index and slices

String slicing (ex_3.2.py)

Create a program that contains the following string:

phrase = ‘Pony stash token’

- Print the string slice ‘stash’ using positive indices
- Print the string slice ‘Pony’ using negative indices
- Print the phrase in reverse
- How to get the string ‘Python’?

print(“Let’s take a 15 min break!”)

Strings are Immutable

- Since strings are immutable, you can not change or re-assign characters within strings

```
>>> name = 'python'
>>> name[0] = 'b'        Type Error
```

- However, you can create a new string

```
>>> new_name = 'b' + name[1:]
>>> print(new_name)
bython
```

String Length

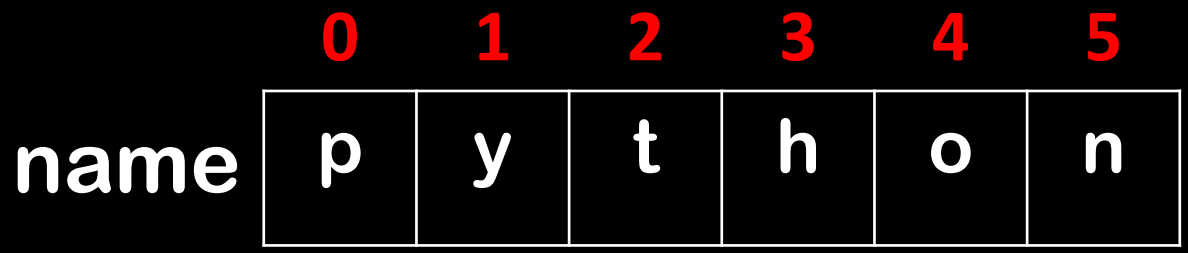
- Python has a built-in function that can be used to get the string length

- len(string) returns the total count of characters in the string

```
>>> name = 'python'
>>> length = len(name)
>>> print(length)
6
>>> print(name[length])
```

IndexError

String Length



- There are 6 characters in name, but index [6] does not exists!

```
>>> name = 'python'
>>> length = len(name)
>>> print(name[length-1])
n
```

Strings Comparison

- You can compare strings by using the equality operator ==
- It compares the numeric value of the strings
- It returns either True or False

```
>>> print('apple' == 'banana')
False
>>> print('apple' == 'apple')
True
```

Strings Comparison

- Remember: upper and lower case are not the same

```
>>> print('Apple' == 'apple')
False
```

Strings Comparison

- You can also check if strings are not equal using != operator

```
>>> print('Apple' != 'apple')
True
```

The in Operator

- The in operator is a operator that checks existence of a string in another string:

```
>>> print('p' in 'apple')
True
>>> print('ple' in 'apple')
True
```

String Methods

String Methods

- Python has a number of useful string methods that can help manipulate strings:
  - Change letters/sentences case
  - Replace characters
  - Search for characters
  - Count characters
  - etc.

String Methods

- Since strings are objects in Python, they use the dot (.) notation

  object.method(argument)

- All string methods return a value! They do not change the string!

- Example:

```
>>> name = "python"
>>> upper_name = name.upper()
>>> print(upper_name)
PYTHON
```

String Methods

Method syntax	Method function
.upper()	Returns a copy of the string converted to upper case
.lower()	Returns a copy of the string converted to lower case
.capitalize()	Returns a copy of the string with the first character capitalized
.title()	Returns a copy of the string as title cased version (first character of every word)
.isupper() .islower()	Returns True if all characters within a string are upper/lower case, and False otherwise

More methods: https://docs.python.org/3/library/stdtypes.html#string-methods

Breakout session III:
String slicing and methods

String methods (ex_3.3.py)

Write a program that requests the following from the user:

1. First and last name using one input() statement
   - in lower case and separated by a space
2. Birthday date represented by an 8-digit integer
   e.g., 19530512 (May 12 1953)

The program should output

1. the first and the last name (both should be capitalized)
2. the string “was born on”
3. the birthday in the standard format DD/MM/YYYY

Example:

Homer Simpson was born on 12/05/1953

Next Class

- More string methods
- Sequences
- Lists
- List methods