

## Mutability

---

## Announcements

## Objects

(Demo)

## Objects

---

- Objects represent information
- They consist of data and behavior, bundled together to create abstractions
- Objects can represent things, but also properties, interactions, & processes
- A type of object is called a class; **classes** are first-class values in Python
- Object-oriented programming:
  - A metaphor for organizing large programs
  - Special syntax that can improve the composition of programs
- In Python, every value is an object
  - All **objects** have **attributes**
  - A lot of data manipulation happens through object **methods**
  - Functions do one thing; objects do many related things

## Example: Strings

(Demo)

## Representing Strings: the ASCII Standard

American Standard Code for Information Interchange

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0 0 1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0 1 0	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
0 1 1	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
1 0 0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1 0 1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
1 1 0	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
1 1 1	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

8 rows: 3 bits  
16 columns: 4 bits

- Layout was chosen to support sorting by character code
- Rows indexed 2-5 are a useful 6-bit (64 element) subset
- Control characters were designed for transmission

(Demo)

## Representing Strings: the Unicode Standard

- 137,994 characters in Unicode 12.1
- 150 scripts (organized)
- Enumeration of character properties, such as case
- Supports bidirectional display order
- A canonical name for every character

8071	8072	8073	8074	8075	8076	8077	8078
聳	聳	聳	聳	聳	聳	聳	聳
健	腓	腓	腓	腓	腓	腓	腓
8171	8172	8173	8174	8175	8176	8177	8178
艷	艷	艷	艷	艷	艷	艷	艷
8271	8272	8273	8274	8275	8276	8277	8278
莖	莖	莖	莖	莖	莖	莖	莖
8371	8372	8373	8374	8375	8376	8377	8378
葱	葱	葱	葱	葱	葱	葱	葱

[http://ian-alfect.com/unicode\\_chart/unicode-chart-chinese.jpg](http://ian-alfect.com/unicode_chart/unicode-chart-chinese.jpg)

LATIN CAPITAL LETTER A

DIE FACE-6

EIGHTH NOTE



(Demo)

## Mutation Operations





## Mutable Functions

## A Function with Behavior That Varies Over Time

Let's model a bank account that has a balance of \$100

```

>>> withdraw = make_withdraw_list(100)
>>> withdraw(25)
75
>>> withdraw(25)
50
>>> withdraw(60)
'Insufficient funds'
>>> withdraw(15)
35
    
```

In a (mutable) list referenced in the parent frame of the function

Return value: remaining balance

Argument: amount to withdraw

Different return value!

Second withdrawal of the same amount

Where's this balance stored?

## Mutable Values & Persistent Local State

