py4kids (https://github.com/wgong/py4kids)

Python for Math - Number, Operation, Variable

In this lesson, we learn how python works with

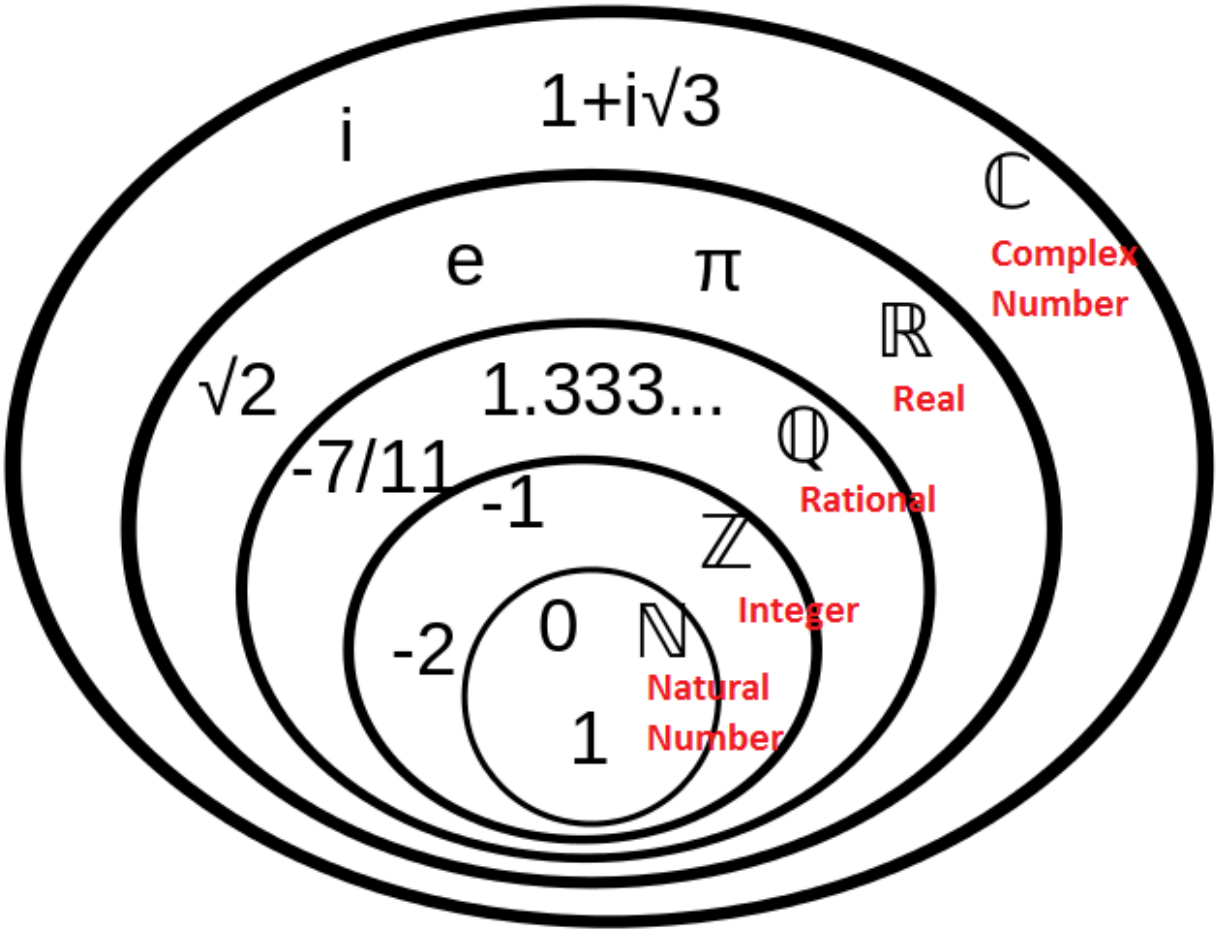
- numbers,
- operations
- variables.

In [1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

Out[1]:

Number System

https://www.wikiwand.com/en/Number (https://www.wikiwand.com/en/Number)



Arithmetic Operations

Symbol	Operation
+	Addition
-	Subtraction
/	division
%	mod
*	multiplication
//	floor division
**	to the power of

Integer

{... -3, -2, -1, 0, 1, 2, 3, ...}

In [2]:
```
1+2+3
```

Out[2]: 6

In [3]:
```
5-5
```

Out[3]: 0

In [4]:
```
4*4
```

Out[4]: 16

In [5]:
```
12 / 3
```

Out[5]: 4.0

In [6]:
```
1 / 2
```

Out[6]: 0.5

In [7]:
```
11 % 10  # remainder
```

Out[7]: 1

In [8]:
```
3//2   # quotient
```

Out[8]: 1

In [9]:
```
3 ** 2  # 3 squared
```

Out[9]: 9

Rational Numbers

{1/3, 7/11, ...}

In [10]:
```
1/3
```

Out[10]: 0.3333333333333333

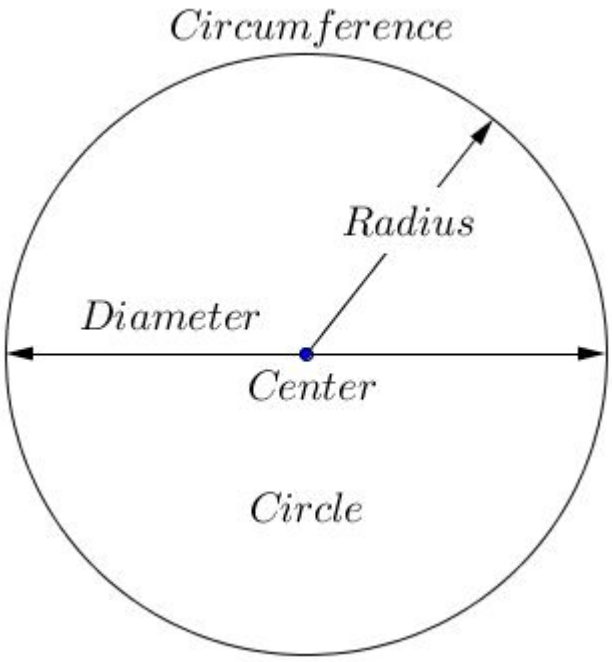
In [11]:
```
-7/11
```

Out[11]: -0.6363636363636364

Irrational Numbers

$\{ \pi, e, \sqrt{2}, ... \}$

In [12]:
```
import math
```



For a circle,

$$\pi = \frac{c}{d}$$
$$A = \pi r^2$$

c - circumference, d - dimeter, r - radius, A - area

In [13]:
```
math.pi
```

Out[13]: 3.141592653589793

In [14]:
```
math.e
```

Out[14]: 2.718281828459045

What is square root of 2?

In [15]:
```
math.sqrt(2)
```

Out[15]: 1.4142135623730951

Do you know the Golden Ratio?

$$R = \frac{1 + \sqrt{5}}{2}$$

In [16]:
```
(1+math.sqrt(5))/2.0
```

Out[16]: 1.618033988749895

Floating-point number

Rational and Irrational numbers together are called Real Numbers.

In computer language, real number is often called Floating-point number, or double number (with more precision). Float/Double numbers carry a decimal point.

What Every Computer Scientist Should Know About Floating-Point Arithmetic (http://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html)

In [17]:
```
print(0.11 + 0.22 - 2*0.11)
```

```
0.11000000000000001
```

Question: why the above answer is not exactly 0.11?

Decimal - when you need more precision

- decimal module (https://docs.python.org/3/library/decimal.html)

In [18]:
```
from decimal import Decimal, getcontext
getcontext().prec = 20
result = 3 * Decimal(0.1)
print(type(result))
```

```
<class 'decimal.Decimal'>
```

In [19]:
```
print(3 * Decimal(0.1))
```

```
0.30000000000000001665
```

In [20]:
```
print(3 * 0.1)
```

```
0.30000000000000004
```

In [21]:
```
import math
print(Decimal(math.pi))
```

```
3.1415926535897931159979634685441851615905761718750
```

In [22]:
```
print(math.pi)
```

```
3.141592653589793
```

Scientific Notation

- useful for very small and very large numbers

In [23]:
```
a_biggy = 1.23e123
```

In [24]:
```
print(a_biggy)
```

```
1.23e+123
```

In [25]:
```
a_biggy**2
```

Out[25]:
```
1.5129000000000001e+246
```

In [26]:
```
a_tiny = 0.9999e-234
```

In [27]:
```
a_tiny**3
```

Out[27]:
```
0.0
```

In [28]:
```
math.sqrt(a_tiny)
```

Out[28]:
```
9.999499987499374e-118
```

A few fundamental constants

speed of light (https://www.wikiwand.com/en/Speed_of_light) - Nothing can fly faster than light

$c = 3.00 \times 10^8 m/s$

In [29]:
```
speed_of_light = 2.99792458E8
print(speed_of_light)
```

```
299792458.0
```

Gravitational constant (https://www.wikiwand.com/en/Gravitational_constant) - Newton's law of gravitation : Every matter is attactive to another matter.

$G = 6.67408 \times 10^{-11} m^3 \cdot kg^{-1} \cdot s^{-2}$

In [30]:
```
G_constant = 6.67E-11
print(G_constant)
```

```
6.67e-11
```

Planck constant (https://www.wikiwand.com/en/Planck_constant) - the unit of quantum universe

$h = 6.62607004081 \times 10^{-34} J \cdot s$

In [31]:
```
h = 6.626E-34
print(h)
```

```
6.626e-34
```

Planck length (https://www.wikiwand.com/en/Planck_length) - when a ruler is so tiny, no one knows what you are measuring

$l_p = 1.616229(38) \times 10^{-35} m$

In [32]:
```
l_p = 1.616E-35
print(l_p)
```

```
1.616e-35
```

Imaginary and Complex Numbers

$$i = \sqrt{-1}$$
$$c = 3 + 4i$$

In [33]:
```
complex(3,4)
```

Out[33]: (3+4j)

In [34]:
```
import math
```

In [35]:
```
math.sqrt(abs(complex(3,4)*complex(3,-4)))
```

Out[35]: 5.0

Euler Equation

$$e^{i\pi} + 1 = 0$$

is claimed to be the most elegant math equation. Why?

It combines the most basic math symbols: $\{0, 1, e, \pi, i, +, =\}$ into one simple equation.

In [36]:
```
import cmath
```

In [37]:
```
abs(cmath.exp(complex(0,1)*cmath.pi)+1)
```

Out[37]: 1.2246467991473532e-16

ASCII - how computer recognizes / represents numbers

https://www.wikiwand.com/en/ASCII (https://www.wikiwand.com/en/ASCII)

In the digital world, everything is made up of bits: (0,1)

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	Start of Header	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	Start of Text	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	End of Text	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	End of Transmission	36	24	044	&#036;	$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enquiry	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Acknowledgment	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Backspace	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	Horizontal Tab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	Line feed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	Vertical Tab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	Form feed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	Carriage return	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	Shift Out	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	Shift In	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	Data Link Escape	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	Device Control 1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	Device Control 2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	Device Control 3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	Device Control 4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	Negative Ack.	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Synchronous idle	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	End of Trans. Block	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Cancel	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	End of Medium	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Substitute	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Escape	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	File Separator	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	|
29	1D	035	Group Separator	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	Record Separator	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	Unit Separator	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Del

asciitbl.com

hex() - from decimal to hexadecimal (4-bit binary)

In [38]:
```
hex(49)
```

Out[38]:
```
'0x31'
```

oct() - from decimal to octal (3-bit binary)

In [39]:
```
oct(49)
```

Out[39]:
```
'0o61'
```

chr() - from integer to ASCII char

In [40]:
```
chr(65), chr(122)
```

Out[40]:
```
('A', 'z')
```

int() - from binary to decimal

In [41]:
```
int('0x31',16)
```

Out[41]:
```
49
```

In [42]:
```
int('0o61',8)
```

Out[42]:
```
49
```

In [43]:
```
int('1010',2)
```

Out[43]: 10

```
ord() - from ASCII to decimal
```

In [44]:
```
ord('b')
```

Out[44]: 98

Python built-in functions

https://docs.python.org/3/library/functions.html (https://docs.python.org/3/library/functions.html)

		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

What is variable?

In algebra, we use a string label to represent number.

In computer, variable stores information of various types.

Computer language speaks of numbers, variables, operations.

In [45]:
```
x = 3
y = 10000000000000000000000000.1
```

In [46]:
```
print(x)
```

```
3
```

In [47]:
```
print(y)
```

```
1e+26
```

In [48]:
```
z = x*y
print(z)
```

```
3.0000000000000003e+26
```

In [49]:
```
z1 = y/3
print(z1)
```

```
3.3333333333333335e+25
```

In [50]:
```
my_big_number = y**3
```

In [51]:
```
print(my_big_number)
```

```
1.0000000000000002e+78
```

In [52]:
```
type(x)
```

Out[52]:
```
int
```

In [53]:
```
type(y)
```

Out[53]:
```
float
```

In [54]:
```
c = complex(3,4)
```

In [55]:
```
type(c)
```

Out[55]:
```
complex
```

In [56]:
```
abs(c)
```

Out[56]:
```
5.0
```

In [57]:
```
math.sqrt(c)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-57-22033c69ef47> in <module>()
----> 1 math.sqrt(c)

TypeError: can't convert complex to float
```

In [58]:
```
cmath.sqrt(c)
```

Out[58]:
```
(2+1j)
```

Naming Rules

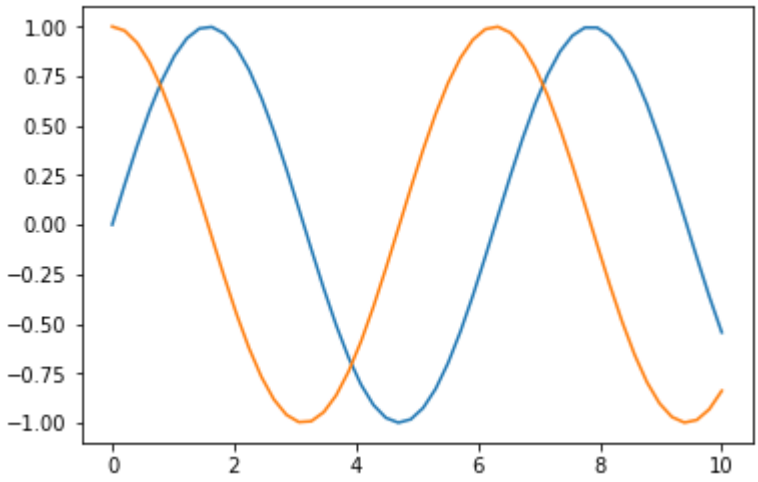
- Name your variable nicely
- Variables can only contain letters, numbers, and underscores. Variable names can start with a letter or an underscore, but can not start with a number.
- Spaces are not allowed in variable names, so we use underscores instead of spaces. For example, use student_name instead of "student name".
- You cannot use Python keywords (http://docs.python.org/3/reference/lexical_analysis.html#keywords) as variable names.
- Variable names should be descriptive, without being too long. For example mc_wheels is better than just "wheels", and number_of_wheels_on_a_motorycle.
- Be careful about using the lowercase letter l and the uppercase letter O in places where they could be confused with the numbers 1 and 0.

How about graph?

In [59]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

In [60]:
```
# plot sin(x), cos(x)
x = np.linspace(0,10)
y1 = np.sin(x)
y2 = np.cos(x)
plt.plot(x,y1,x,y2)
```

Out[60]:
```
[<matplotlib.lines.Line2D at 0x7f06eb8>,
 <matplotlib.lines.Line2D at 0x7f10908>]
```



Fun and Fancy Math

In [61]:
```
import sympy
sympy.init_printing(use_latex='mathjax')
x = sympy.symbols('x')
```

In [62]:
```
from IPython.display import display
from IPython.html.widgets import interact

@interact
def factorit(n=10):
    expr = x**n-1
    display(expr)
    print('=')
    display(sympy.factor(expr))
```

n 10

$$x^{10} - 1$$

=

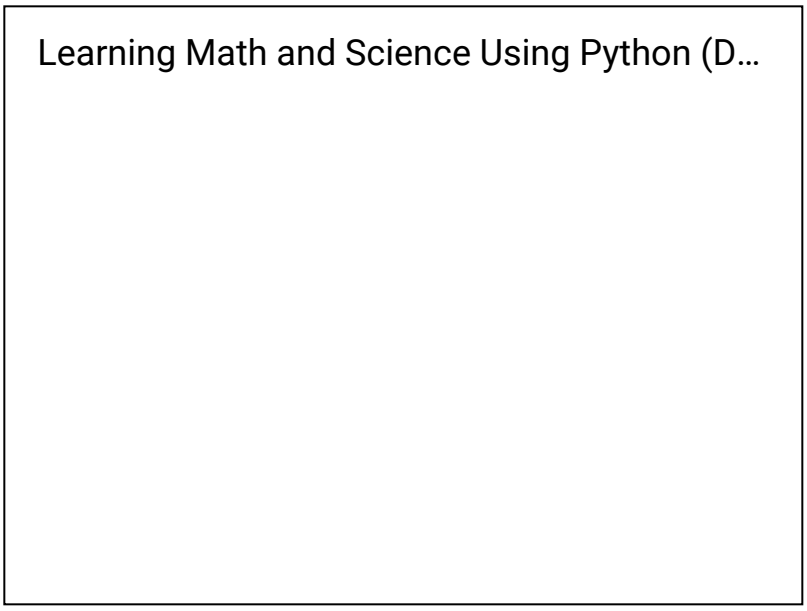
$$(x - 1)(x + 1)\left(x^4 - x^3 + x^2 - x + 1\right)\left(x^4 + x^3 + x^2 + x + 1\right)$$

Learn Math using Python - YouTube

In [63]:
```
from IPython.display import Image, YouTubeVideo
```

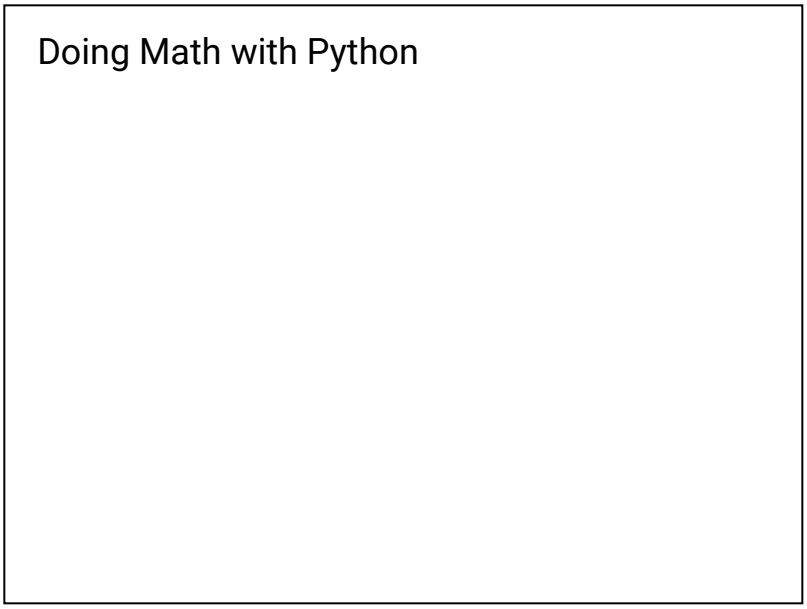
In [64]:
```
YouTubeVideo('HfvQ6O7Di0g')
```

Out[64]:



In [65]:
```
YouTubeVideo('XJOt4QQgx0A')
```

Out[65]:



In [ ]:
```

```