
CODE BREAKERS
PYTHON & MACHINE LEARNING BY MANIK GHAI & SURENDRA JAISWAL

PYTHON

- Introduction to coding
- Python Basics
- Python Modules
- Web Development using Python
- Machine Learning using Python

Manik Ghai, Director
Credit Suisse Singapore IT

Achievements

Hosted Hackathon for Credit Suisse for 330 people, 4 years in a row

Education

Master in computer application. Certificate in Finance and Insurance .

Contact Info

Manik.ghai@gmail.com

Hobbies

Tennis, travelling and Running

Skills and Language

Java, Microsoft Dot Net, Python, Data Analytics, Architecture

Work Experience

16 years of IT and banking domain experience

Surendra Jaiswal, AVP
Credit Suisse IT

Achievements

Completed half marathon in less than 3 hours.

Education

Master in computer application. Certificate in banking from institute of banking and finance

Contact Info

jaiswal.surendra@gmail.com

Hobbies

Plying Badminton and Running

Skills and Language

Java, Microsoft Dot Net, Python, Chatbot, Machine Learning

Work Experience

14 years of IT and banking domain experience

INTRODUCTION TO CODING

- What is code and what is coding
- What are core parts of starting coding
- Pseudo code

PYTHON BASICS

- Programming basics
- Expressions, Variables
- Print, Input
- If/If else, For loop, While
- Logical operators
- Data Types
- Arithmetic operators

PYTHON MODULES

- Turtle- "**Turtle**" is a **python** drawing board, which lets you command a **turtle** to draw all over it!
- Tkinter- Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI.

WEB DEVELOPMENT

- **Django-** Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development

MACHINE LEARNING-OVERVIEW

- What is AI/Machine Learning
- Why use machine learning?
- Applications of machine learning
- Types of machine learning

MACHINE LEARNING-IMPLEMENTATION

- Anaconda-Anaconda is the standard platform for Python data science, leading in open source innovation for machine learning. It includes over 300 most popular python packages
- NumPy-Package for scientific computing
- Pandas-Matplotlib- Visualization with powerful, flexible and easy to use open source data analysis and manipulation tool
- Linear regression-**Linear regression** is a way to explain the relationship between a dependent variable and one or more explanatory variables using a straight line. **Linear regression** can be used to fit a predictive model to a set of observed values (data). This is useful, if the goal is prediction, or forecasting, or reduction. For example predicting house price.

CODE BREAKERS

PYTHON & MACHINE LEARNING

DAY 1

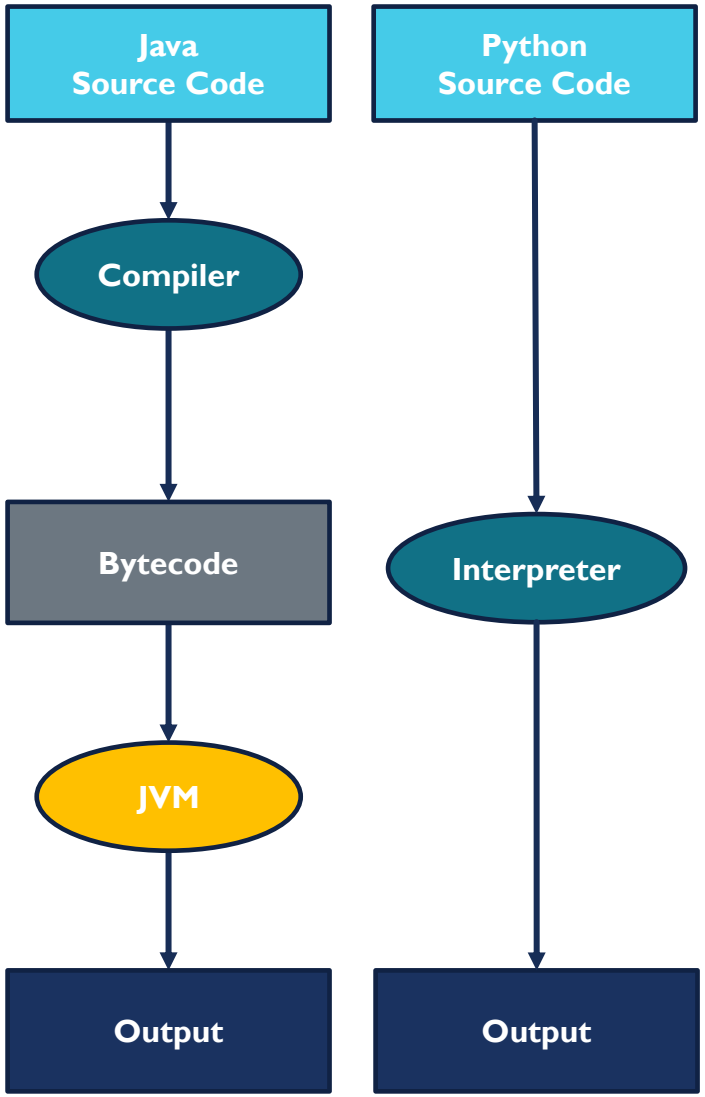
- What is code and what is coding
- What are core parts of starting coding
- Pseudo code

MODULE 1 INTRODUCTION TO CODING

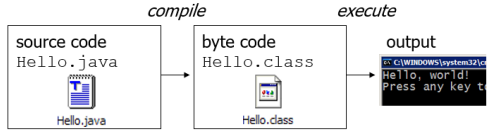
WHAT IS CODE AND WHAT IS CODING

1. Computer code or program code is the set of instructions forming a computer program which is executed by a computer. This source code is translated into machine code by a compiler or interpreter so that the computer can execute it to perform its tasks.

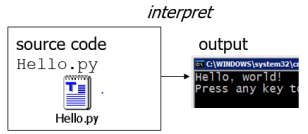
2. Coding or Programming language is a set of syntax rules that define how code should be written and formatted. Thousands of different programming languages make it possible for us to create computer software, apps and websites.



Many languages require you to *compile* (translate) your program into a form that the machine understands.



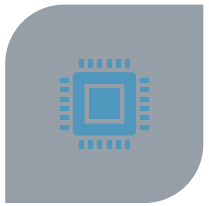
Python is instead directly *interpreted* into machine instructions.



WHAT ARE CORE PARTS OF STARTING CODING

LOGICAL THINKING

PSEUDO CODE



IDENTIFYING WHICH PROGRAMMING LANGUAGE TO CHOOSE

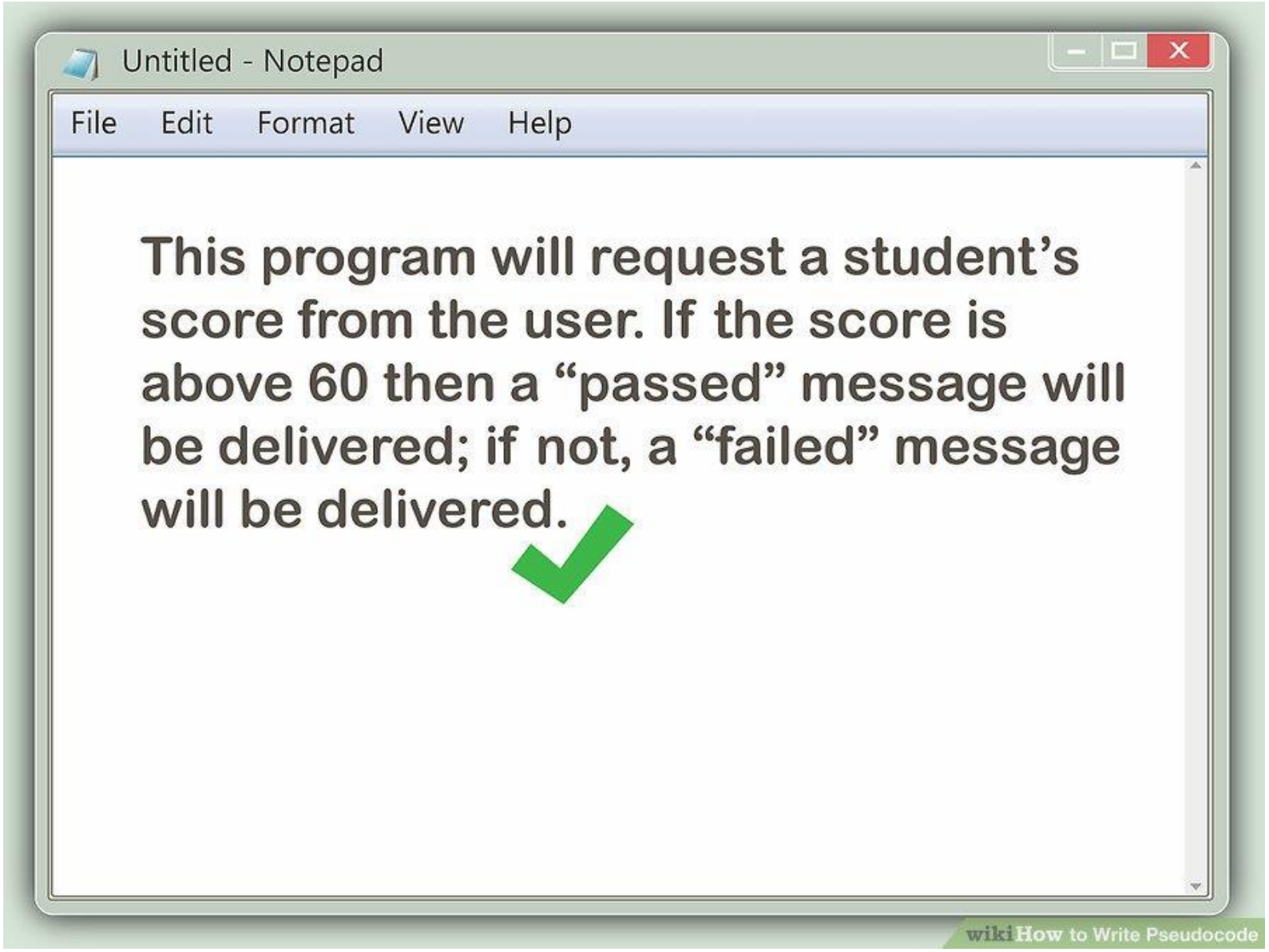
IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

LEARNING THE SYNTAX

COMMENTS IN YOUR CODE

COLLABORATION AND SOURCE CODE REPOSITORIES

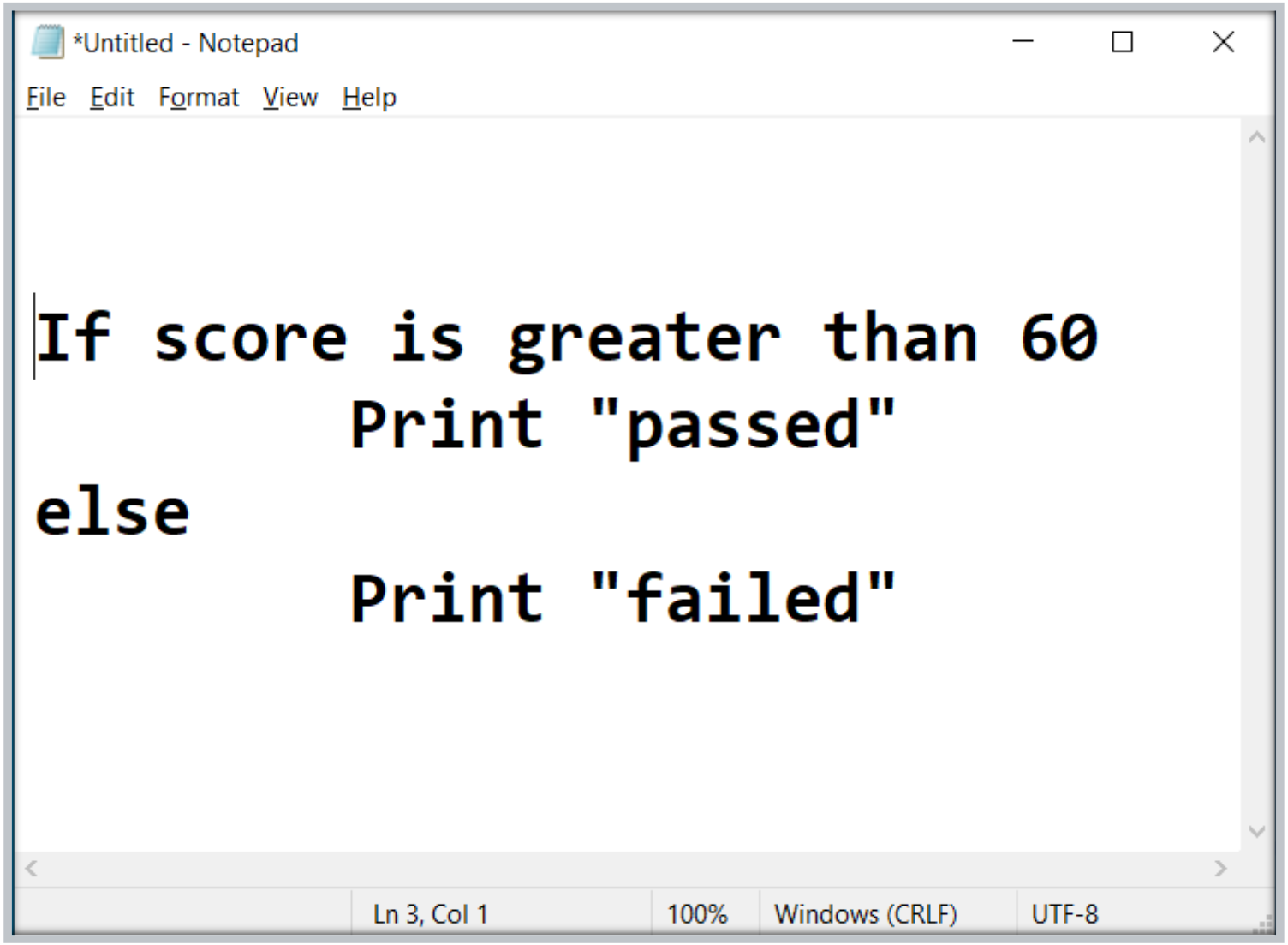
PROBLEM STATEMENT



LOGICAL THINKING & PSEUDO CODE

What is pseudocode?

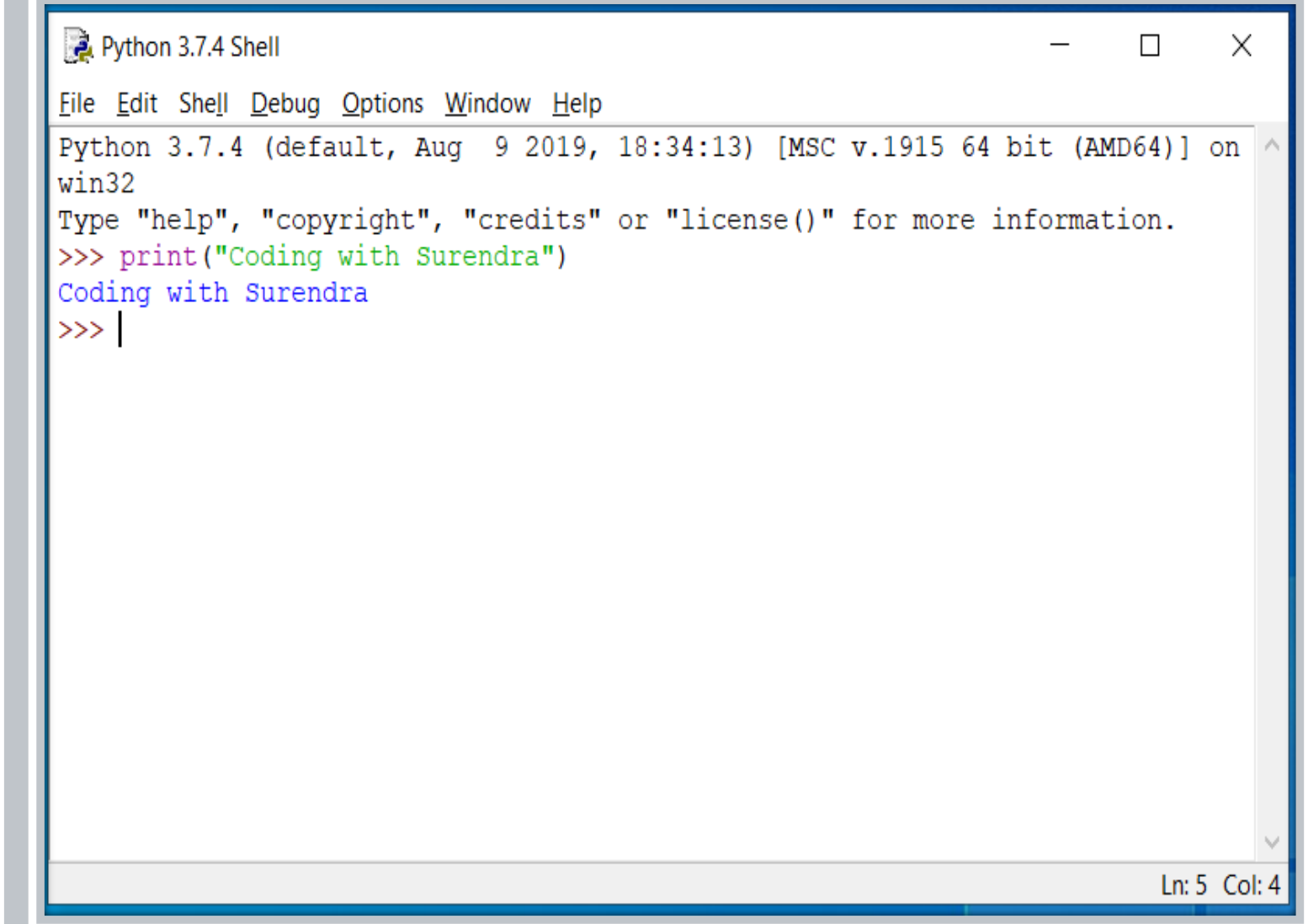
- Pseudocode is step-by-step written outline of your
- Code that you can gradually transcribe into the programming language.



IDE(INTEGRATED DEVELOPMENT ENVIRONMENT)

What is IDE?

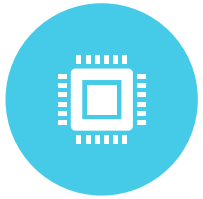
- An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger



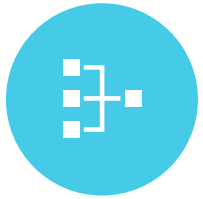
- Python Basics

MODULE 2

PYTHON BASICS



Programming basics

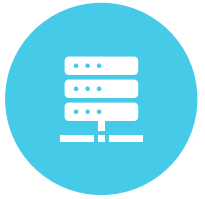


Expressions, Variables

Print, Input

If/If else, For loop, While

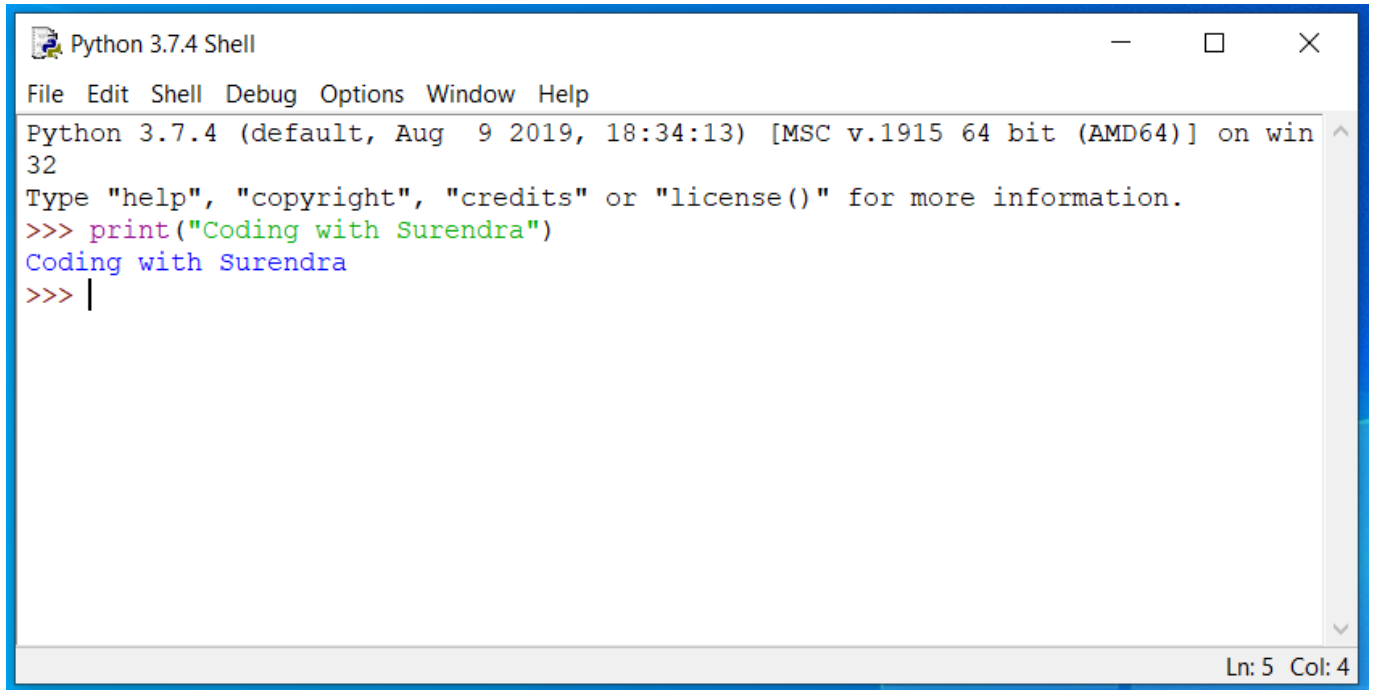
Logical operators



Data Types

PROGRAMMING BASICS

- **code** or **source code**: The sequence of instructions in a program.
- **syntax**: The set of legal structures and commands that can be used in a particular programming language.
- **output**: The messages printed to the user by a program.
- **console**: The text box onto which output is printed.



EXPRESSIONS

- A data value or set of operations to compute a value.
- Examples:

```
1 + 4 * 3
```

```
13
```

VARIABLES

A named piece of memory that can store a value.

- Usage:
  - Compute an expression's result,
  - store that result into a variable,
  - and use that variable later in the program.

```
name=value
```

Example: Stores a value into a variable.

```
y=5
```

```
gpa=3.14
```

A variable that has been given a value can be used in expressions.

```
y*3
```

PRINT

```
print : Produces text output on the console
```

```
print("Message")
```

```
>>> print("Hello")
```


```
print(Expression)
```

```
>>> a=10
>>> print(a)
```


```
print(Item1, Item2, ..., ItemN)
```

```
>>> a="Gold Price"
>>> b=7540
>>> print(a,b)
```

ARITHMETIC OPERATORS

addition

```
>>> 3 + 5
8
```

subtraction

```
>>> 10 - 5
5
```

multiplication

```
>>> 3 * 5
15
```

division

```
>>> 30 / 5
6
```

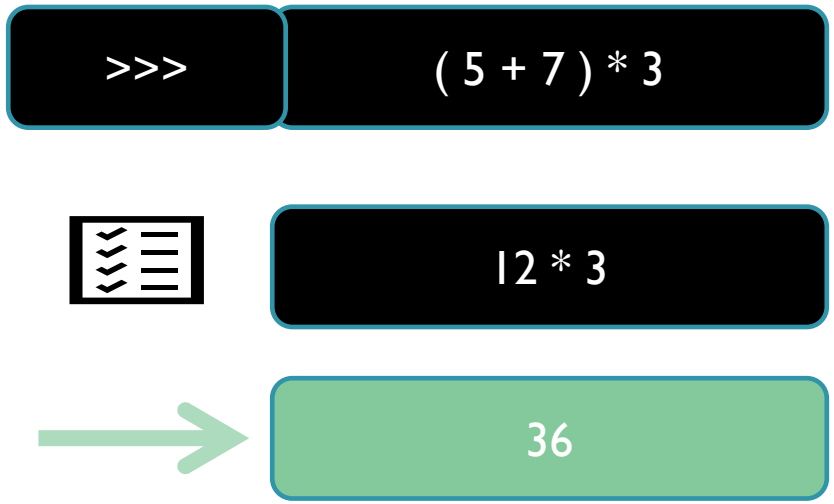
exponent

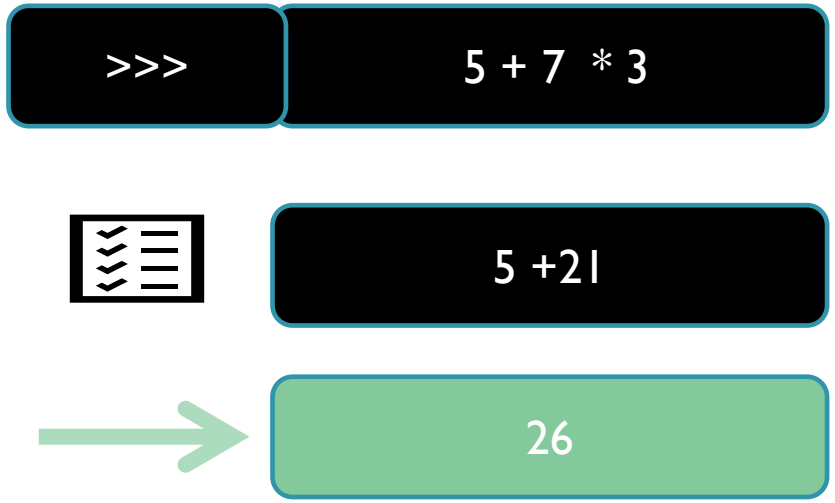
```
>>> 2 ** 3
8
```

modulus

```
>>> 32 % 5
2
```

ORDER OF OPERATIONS





PEMDAS: **P**arentheses **E**xponent **M**ultiplication **D**ivision **A**ddition **S**ubtraction

DATA TYPES

int

```
>>> a=35
```

35

float

```
>>> b=30.5
```

30.5

bool

```
>>> has_passed=True
>>> print(has_passed)
```

True

str

```
>>> string_1 = "Hello"
>>> string_2 = "World"
>>> print (string_1 + string_2)
```

HelloWorld

INPUT

`input : Reads a number from user input.` You can assign (store) the result of `input` into a variable.

Example

```
money= input("How much money you have? ")
print("Your have : ", money)
```

```
How much money you have? 500
Your have :  500
```

EXERCISE-1

Write a Python program that prompts the user for his/her amount of money, then reports how many Xbox the person can afford, and how much more money he/she will need to afford an additional Xbox.

Assume latest xBox one price $ 450

EXERCISE 1-ANSWER

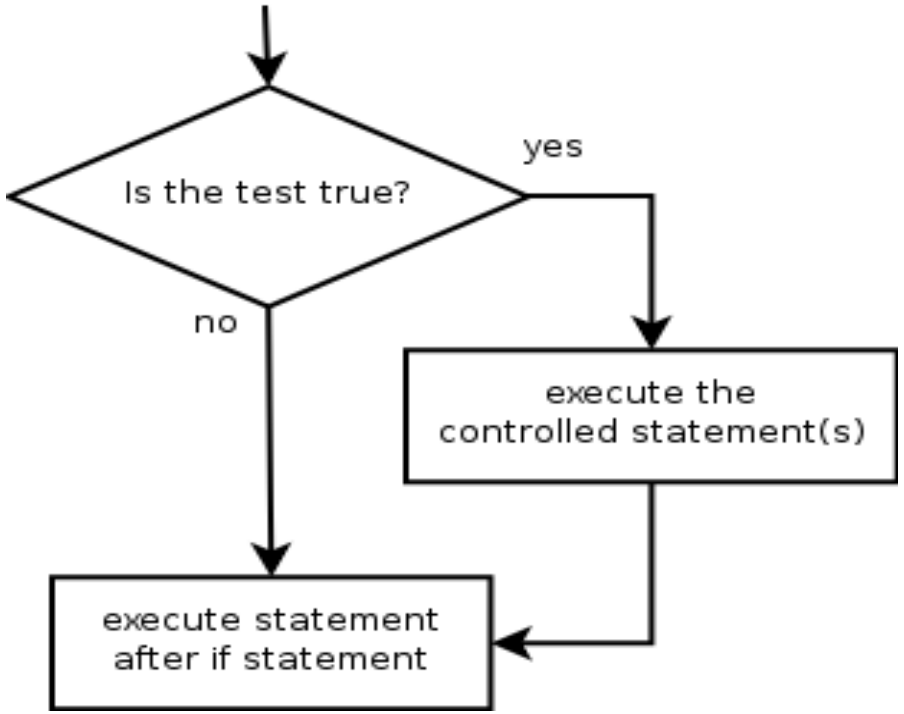
```
money= int(input("How much money you have? "))
print("Your can afford", int(money/450),"xbox")
print("You need ", 450-money%450, " to buy additional one xbox")
```

```
How much money you have? 1050
Your can afford 2 xbox
You need  300  to buy additional one xbox
```

IF

- **`if` statement**: Executes a group of statements only if a certain condition is true. Otherwise, the statements are skipped.
  - Syntax:

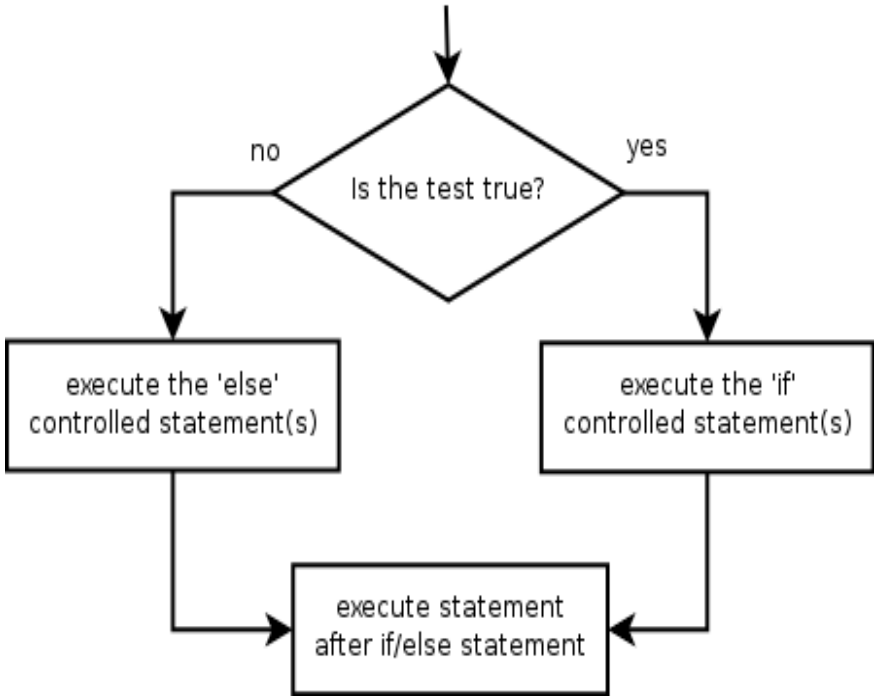
```
if condition :
        statements
```



IF/ELSE

- **`if/else` statement**: Executes one block of statements if a certain condition is True, and a second block of statements if it is False.
  - Syntax:

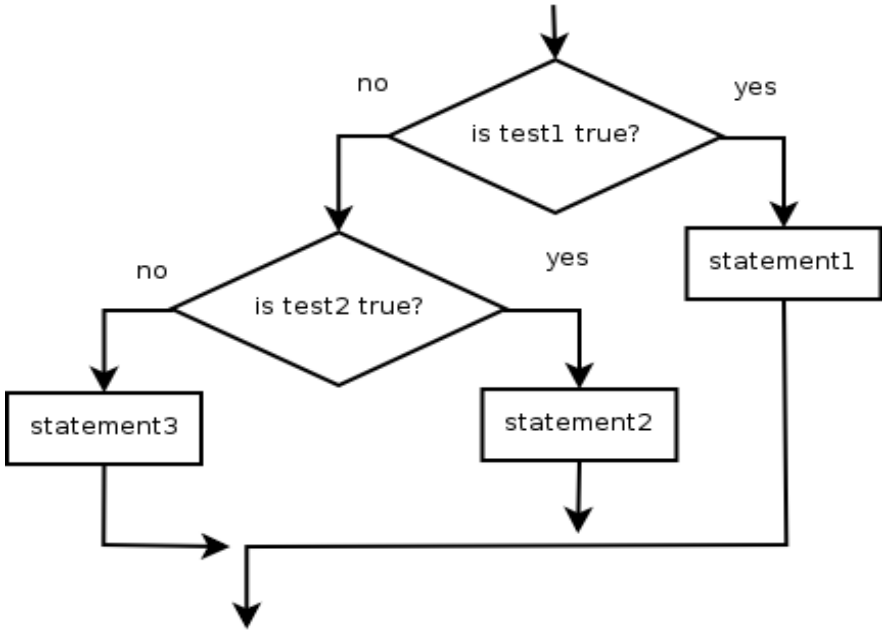
```
if condition :
        statements
else:
        statements
```



IF/ELSE

Multiple conditions can be chained with `elif` ("else if"):

```
if condition :
    statements
elif condition :
    statements
else:
    statements
```



EXAMPLE : IF/ELSE -STUDENT SCORE

```
score= int(input("What is your score?"))
if score> 60:
        print ("passed")
else :
        print ("failed")
```

EXAMPLE : IF/ELIF -ADMISSION APPLICATION

```
gpa = 3.0
if gpa > 4.0:
        print ("Your application is accepted.")
elif gpa > 2.0:
        print ("Your application is on hold.")
else:
        print ("Your application is rejected.")
```

FOR LOOP

`for` loop: Repeats a set of statements over a group of values.
Syntax:

```
for variableName in groupOfValues:
        statements
```

We indent the statements to be repeated with tabs or spaces.

variableName gives a name to each value, so you can refer to it in the ***statements***.

groupOfValues can be a range of integers, specified with the `range` function.

Example:

```
for x in range(1, 6):
 print(x, "squared is", x * x)
```

Output:

```
1 squared is 1
2 squared is 4
3 squared is 9
4 squared is 16
5 squared is 25
```

RANGE

- The `range` function specifies a range of integers:
- `range(`***start***`, `***stop***`)`

  the integers between ***start*** (inclusive) and ***stop*** (exclusive)

- It can also accept a third value specifying the change between values.
- `range(`***start***`, `***stop***`, `***step***`)`

  the integers between ***start*** (inclusive) and ***stop*** (exclusive) by ***step***

```
for x in range(5, 0, -1):
    print(x)
print("Blastoff!")
```

Output:

```
5
4
3
2
1
Blastoff!
```

CUMULATIVE LOOPS

Some loops incrementally compute a value that is initialized outside the loop. This is sometimes called a *cumulative sum*.

```
sum = 0
for i in range(1, 11):
    sum = sum + (i * i)
print("sum of first 10 squares is", sum)

Output:
sum of first 10 squares is 385
```

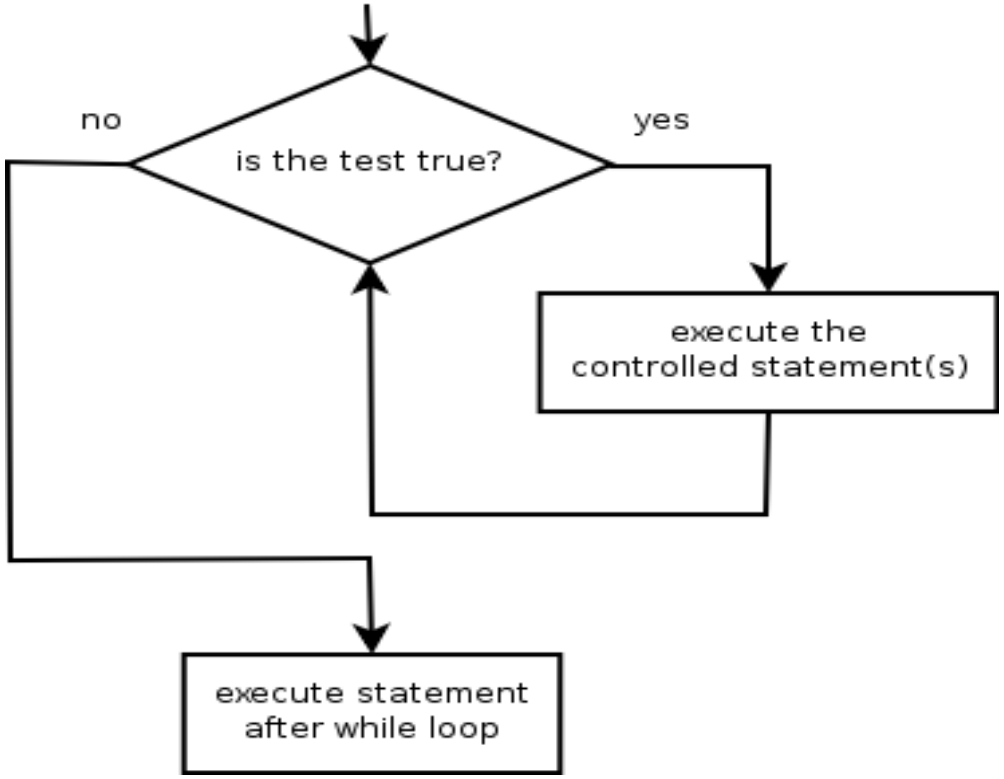
WHILE

- **`while` loop**: Executes a group of statements as long as a condition is True.
- Syntax:

```
while condition:
    statements
```

```
number = 1
while number < 200:
  print(number)
  number = number * 2
```

Output: 1 2 4 8 16 32 64 128



COMMENT

```
>>> print(“Food is very nice") #eat me
```



```
Food is very nice
```

CODE BREAKERS

MACHINE LEARNING

DAY 2

DATA TYPES-TUPLES, LISTS, AND DICTIONARIES

•**Tuples** Tuple is a collection of items of any Python data type, same as the list type. Unlike the list, tuple is immutable. Example: months of the year.

```
>>> months = ('January','February','March','April','May','June',\
'July','August','September','October','November','  December')
```

•**Lists** the list is a collection of items of different data types. It is an ordered sequence of items. A list object contains one or more items, not necessarily of the same type, which are separated by comma and enclosed in square brackets []. Example: Your cats' names

```
>>> cats = ['Tom', 'Snappy', 'Kitty', 'Jessie', 'Chester']
```

•**Dictionary:** A dictionary object is an unordered collection of data in a key:value pair form. A collection of such pairs is enclosed in curly brackets. Example: telephone book.

```
>>> phonebook = {'Andrew Parson':8806336, 'Emily Everett':6784346, 'Peter Power':7658344, \
'Lewis Lame':1122345}
```

DATA TYPES-BUILT-IN FUNCTIONS

A numeric object of one type can be converted in another type.

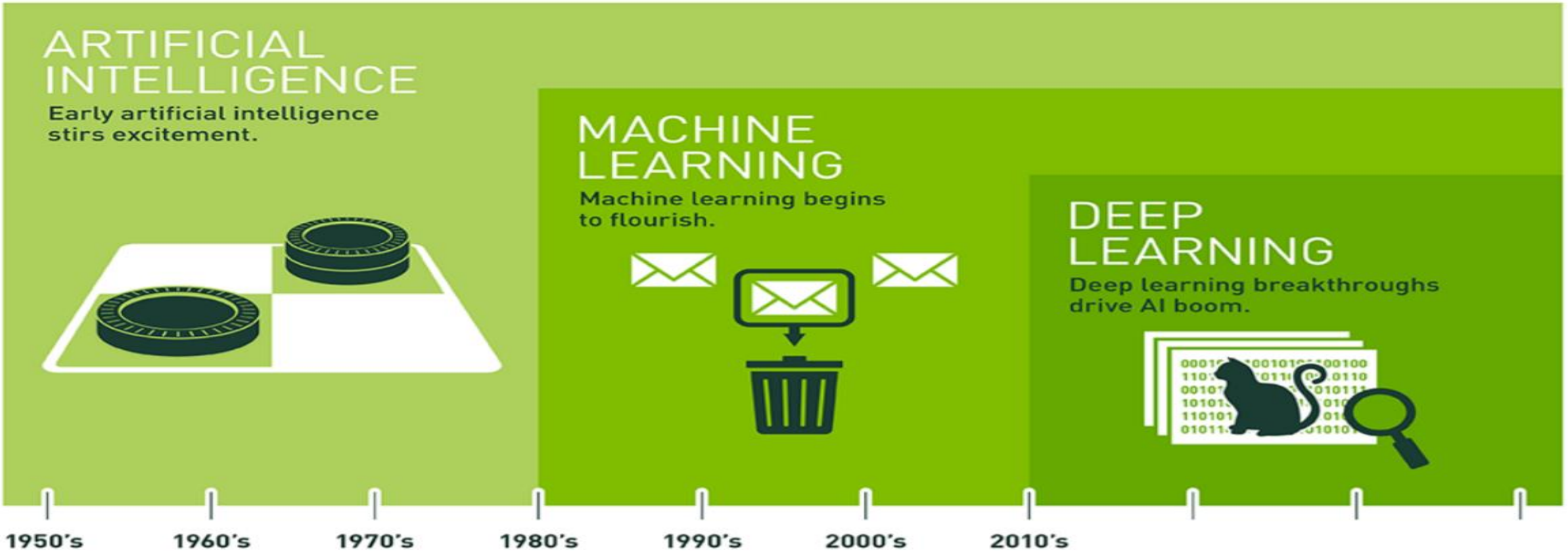
Built-in Function	Description
int	Returns the integer object from a float or a string containing digits.
float	Returns a floating-point number object from a number or string containing digits with decimal point or scientific notation.
complex	Returns a complex number with real and imaginary components.
hex	Converts a decimal integer into a hexadecimal number with 0x prefix.
oct	Converts a decimal integer in an octal representation with 0o prefix.
pow	Returns the power of the specified numbers.
abs	Returns the absolute value of a number without considering its sign.
round	Returns the rounded number.

CODE BREAKERS

MACHINE LEARNING

DAY 10

WHAT IS AI/MACHINE LEARNING?



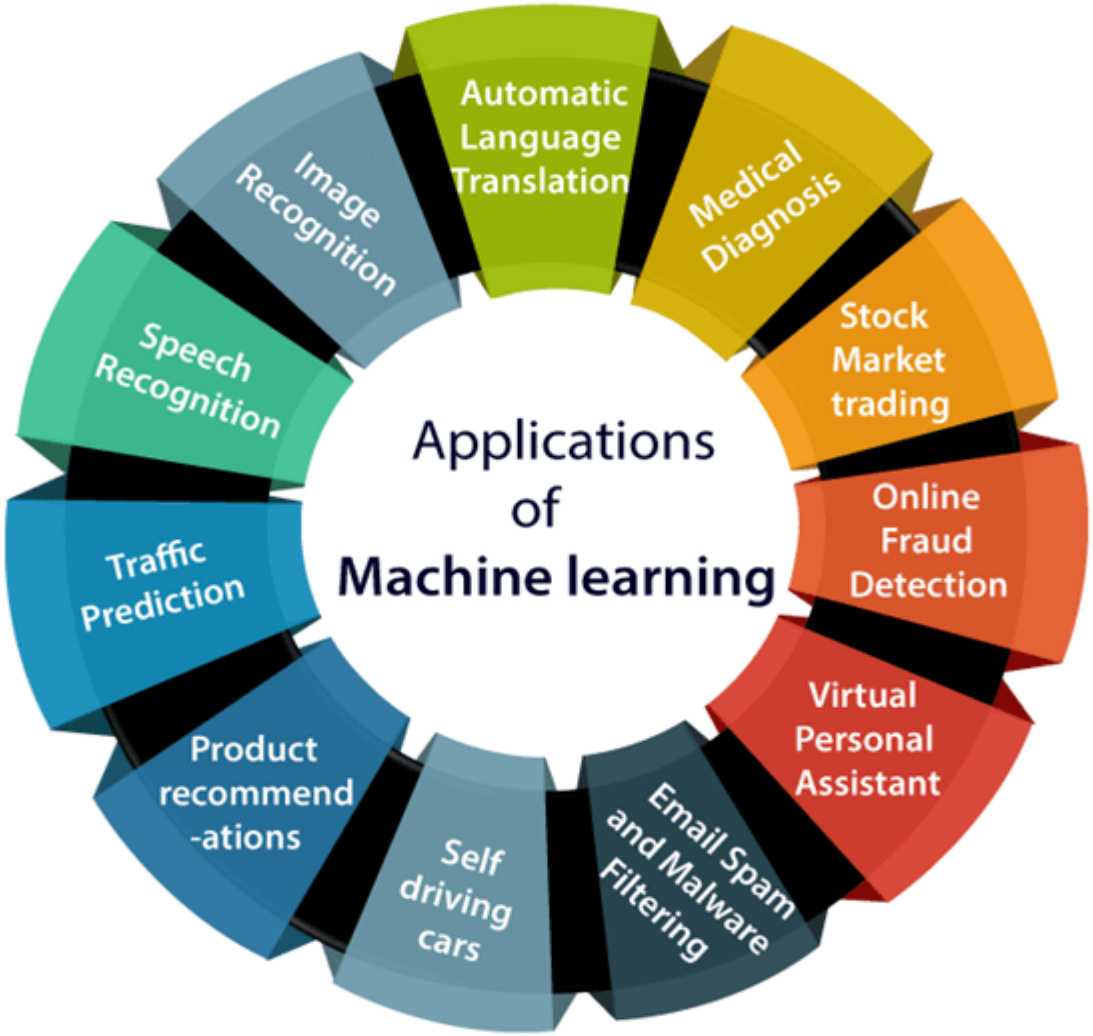
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

WHY USE MACHINE LEARNING?

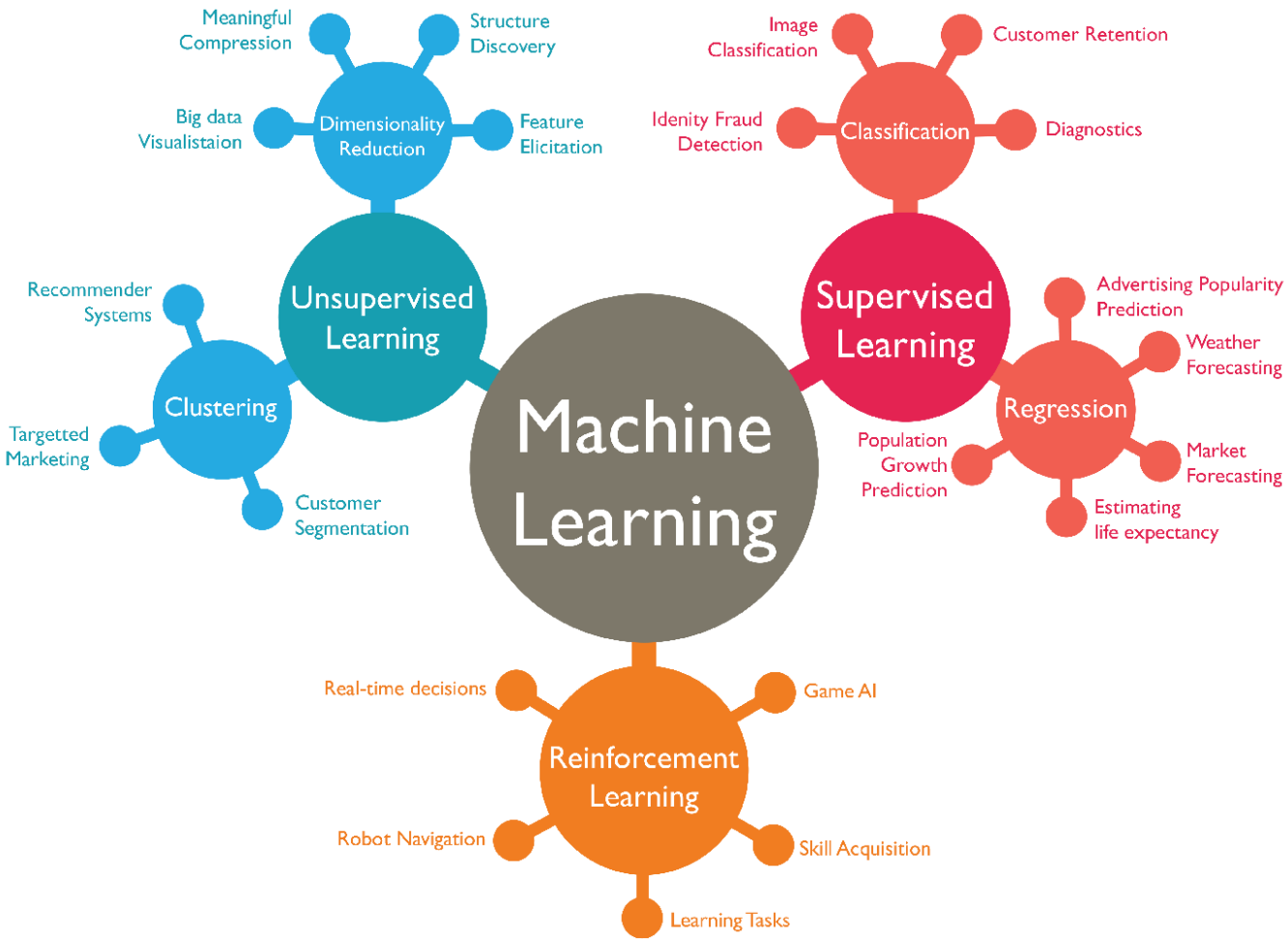
- Problem
  - Can machines do what we (as thinking entities) do?

- Solution
  - Machine Learning uses algorithms that can learn from and make predictions on data
  - Can even outperform humans

APPLICATIONS OF MACHINE LEARNING



TYPES OF MACHINE LEARNING



THANK YOU AND HAPPY CODING

SURENDRA.JAISWAL@CREDIT-SUISSE.COM