

January 2011						
Wk	Mo	Tu	We	Th	Fr	Sa Su
52/5	31				1	2
	1	3	4	5	6	7
2	10	11	12	13	14	15
3	17	18	19	20	21	22
4	24	25	26	27	28	29

Saturday

1

January

2011

Week 52 • 01-364

had capability
to drive
the world.

Lets Upgrade. (AI-ML course.)

6-July-2020

- Data is new oil (When we have data, we can get new features.
analysing data) ↗ ↓

Need of ML →

Humans are most advanced & intelligent lifeform
of this planet at moment.

<input checked="" type="checkbox"/> Excel/Mtc <small>things also do, correctly.</small> <small>never.</small> <small>Seldom req. rest.</small> <small>never</small> <small>But if rows and can't fast</small> <small>(let 2 lakhs, 400 coln)</small>	Humans	
	do things correctly.	
	Tired	
	rest	
	may commit error	

Table
Ex
1
2
3
4
5
6
7
8
9
10
11
12

→ donot require
(ML). Excel can do
things also.

∴ M/c can be trained to do task in faster ways.

ML (1) Make data-driven decision at scale.

ML → use data itself to drive
decision instead of
prog. logic, rules or
code to make
these
decisions.

Sunday

2

Ex. of data new oil 02-363

make my trip
come of
use for
data

Home of
A → B
GPS

via Google,
we get a
data
all together.

there are
more
people who
travel to
same
place to
destination

after
analyzing
data,
they give
a proposal
of Bus



Driving ↓,
effort ↓,
use time for other resource.

Good platform.
Brands
Seats, cost ↓,

3

Monday

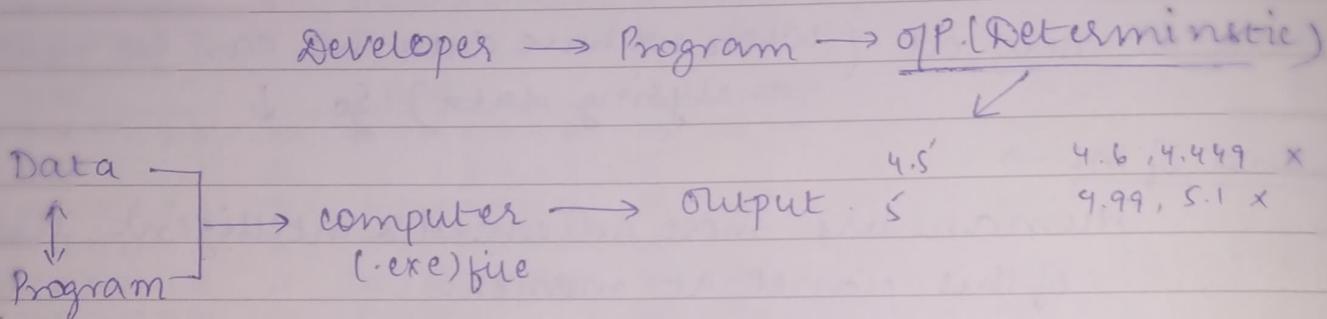
January 2011

Week 1 • 03-362

January							2011	
Wk.	Mo	Tu	We	Th	Fr	Sa	Su	
Sa	31					1	2	
1	3	4	5	6	7	8	9	
2	10	11	12	13	14	15	16	
3	17	18	19	20	21	22	23	
4	24	25	26	27	28	29	30	

complex flowcharts,
if else,
loop coded cycles.

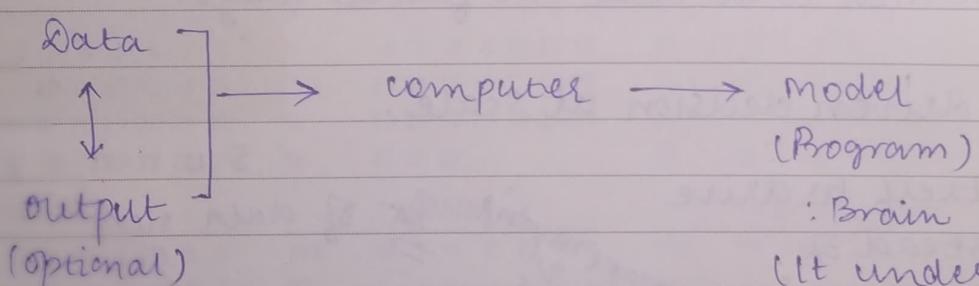
Traditional Programming Program.



when developer write program while executing it generates .exe file. Data is taken from user & gives O.P. here, O.P is deterministic (fixed).

Machine Learning Program.

• O.P : (Predictable (NOT fixed))



: Brain
(It understand the relationship b/w Ques & Ans. & create Brain)

Data is no longer require data.

85-90% → Good model.

February 2011						
Mon	Tue	We	Th	Fr	Sa	Su
5	1	2	3	4	5	6
6	7	8	9	10	11	12
7	14	15	16	17	18	19
8	21	22	23	24	25	26
9	28					

Tuesday

January 2011

4

Week 1 • 04-361

anything which deals with happiness & life should have 100% accuracy → By Google.

- Excel Model X
(data is needed)
- ML model ✓
(data is not needed once
model is trained)

Definition of ML

ML is a field that consists of learning algo. that

1. Improve their performance P
2. At executing some task T
3. Over time with experience E

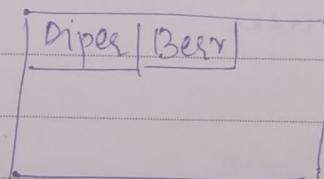
Ex Email classification.

Ques:

Why Domain knowledge is imp.?

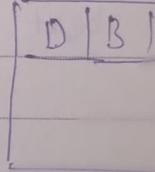
We have to give insight to manager
they should accept or reject → their
decision.

Ex. Walmart USA



sales ↑ 40%.

India - Big Bazaar



Data will
be fined

Age 30 - 35

jo Diper leta hai } zcln.
↓ Beer bi leta hai } zcln.

5

Wednesday

January 2011

Week 1 • 05-360

Teacher is there for
guidance.

January 2011						
Wk	Mo	Tu	We	Th	Fr	Sa Su
52/5	31					1 2
1	3	4	5	6	7	8 9
2	10	11	12	13	14	15 16
3	17	18	19	20	21	22 23
4	24	25	26	27	28	29 30

Supervised Learning.

① Classification

ML: Garbage In
Garbage Out

child remember this

QB	AB
O	c
Δ	T
○	c
△	T
⋮	⋮
○	c

then:

3 sides $\rightarrow T$ no sides $\rightarrow c$ Test: $\Delta, O, \triangle, \square$

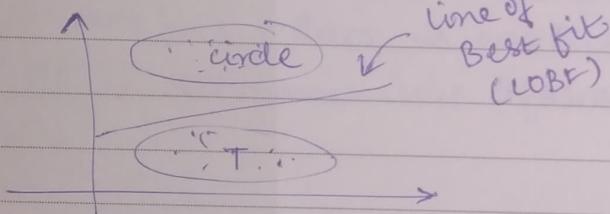
T C T

c/T

wrong test data.

(garbage OP)
Not predict above

MLC :

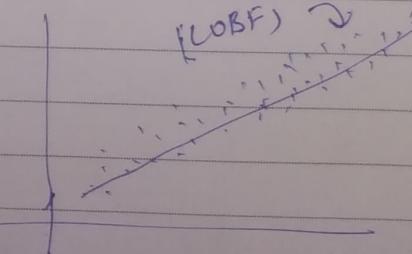
MLC distinguish with line
and calculate error.suppose draws 100 lines, it
takes the line which
has least error & show the
best line to user

② Regression.

involve predⁿ that a real numerical
value is op. instead of class or
category.

(continuous value.)

	Sector	Price
1	10 lakh	
5	50	
9	45	
3	30	
7	30	
18	69	
10	102	



February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa
5		1	2	3	4	5
6	7	8	9	10	11	12
7	14	15	16	17	18	19
8	21	22	23	24	25	26
9	28					

Thursday

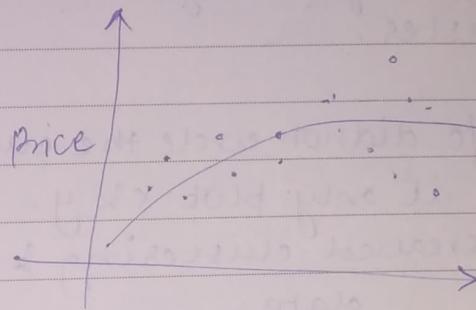
January 2011

6

Week 1 • 06-359

multiple regression. (Polynomial)

when there are multiple features



5 features has 5 Best-fit line.

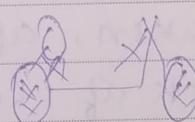
no line can be made from

all 5 lines so generalizing

all we get a curve.

Unsupervised learning.

7-July-2020.

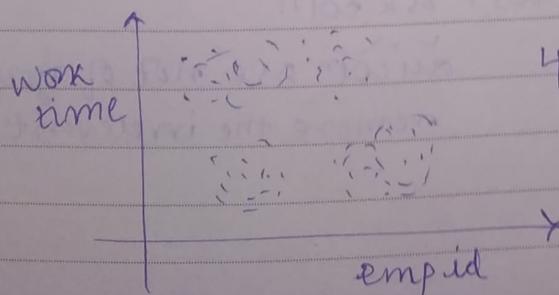


No teacher present.
child is learning by
itself.

1. clustering
2. Dimensionality reduction
3. Anomaly detection
4. Association Rule Mining.

① Clustering.

Ex: total employ \rightarrow 500
time hr: 9 to 5pm (8hrs).



emp. id	work time
1	7
2	6
3	7
4	8
5	9
...	...
500	3

7

Friday

January 2011

Week 1 • 07-358

I want a single cluster but getting 4 clusters. ie something is getting wrong in company. I manually check each cluster.

M/c did not circle the clusters, it only plot x & y.

I created clustering & analyze data.

If there are 4 clusters, 3 or 2 clusters only get after analyzing data.

Owner job:

1. organize a picnic
2. motivation

after few steps taken, after 1 month plot graph again & analyze & check if I had one cluster or close enough. ie owner did a good job.

(2) Dimensionality Reduction.

Features space gets bloated up with a humongous no. of features.

(Select limit no. of features.)

Let you have 50 lakh rows x 65k coln

~ 40% are not of your interest.
remove the irrelevant columns.

January 2011						
Wk	Mo	Tu	We	Th	Fr	Sa
52/5	31					1
1	3	4	5	6	7	8
2	10	11	12	13	14	15
3	17	18	19	20	21	22
4	24	25	26	27	28	29
						30

February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa
5	1	2	3	4	5	6
6	7	8	9	10	11	12
7	14	15	16	17	18	19
8	21	22	23	24	25	26
9	28					

Saturday

January 2011

8

Week 1 • 08-357

mp

My space constraints. (This problem is called - Curse of dimensionality)
This means if you have a lot features, how will you reduce them?

We can reduce either

Manually / Automized (PCA / LDA)

soln.) ① PCA

② Discriminant analysis (linear)

③ k-neighbours.

Ex

Name. RBC WBC Serum Hemoglobin ..

as a data scientist you don't know biologically terms, so we use algo. to do dimensionality reduction for us.

③

Association Rule mining.

Data mining method used to examine and analyze large transactional datasets to find patterns & rules of interest.

- Market Basket Analysis. (often called).
- Used to analyze customer shopping patterns.

Ex

cart

M-30

09-356

9

Before buying you have seen

- You should buy : Glass cover
- Back covers recommendation etc.

Ques.

Why is this an unsupervised method ?

- Applying algo. like apriori algorithm or FP-growth, to detect & predict products or items associated closely with each other & find conditional probabilistic dependencies.

we have no idea which items are strongly related or associated with other, before hand.

10

Monday

January

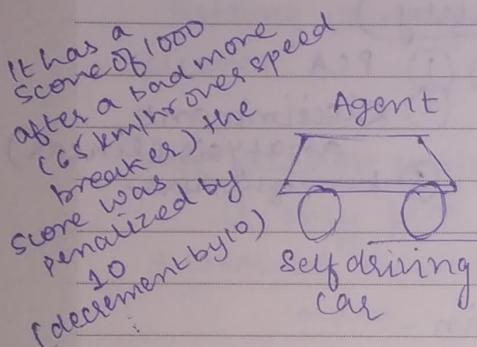
2011

Week 2 • 10-355

this agent works
on task reward
policies. ↪

January							2011
Wk	Mo	Tu	We	Th	Fr	Sa	Su
52/5		31					
1	3	4	5	6	7	1	2
2	10	11	12	13	14	8	9
3	17	18	19	20	21	15	16
4	24	25	26	27	28	29	30

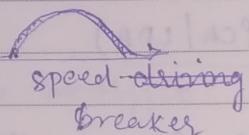
Reinforcement Learning. (We have concept of agent in this)



15 → smoothly goes

36 7. Jumped over.

$$\bullet \text{Score} = 1000 - \underbrace{10}_{\text{penalty}} - \underbrace{10}_{\text{penalty}} + \underbrace{2}_{\text{reward}}$$



Based on action, the agent gets a reward, which could be beneficial or detrimental in form of penalty.

- 9. requires a lot of data to train
- 10. It learns from its own mistakes.
- 11. Updates its current policies & strategies.

Iterative process continues

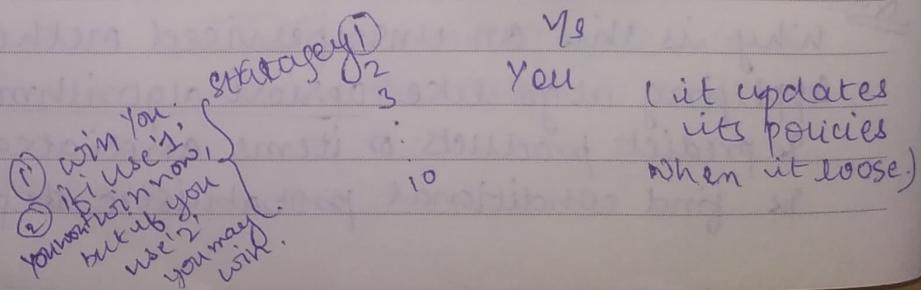
Ex. Google Autonomous car.

Steps of reinforcement learning -

5. (1) Prepare agent with set of policies & strategies.
6. (2) Observe environment & classwork. current state.
7. (3) Select optimal policy & perform action
8. (4) Get corresponding reward (or penalty.)
9. (5) Update policies if needed
10. (6) Repeat steps 2-5 iteratively until agent learns the most optimal policies.

Ex: Real World Ex.

AI enabled chess



based
on
incremental
data
samples

Batch Learning

Online Learning

	February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa	Su
5	1	2	3	4	5	6	
6	7	8	9	10	11	12	13
7	14	15	16	17	18	19	20
8	21	22	23	24	25	26	27
9	28						

Not creating data
but it is abt. Tuesday
Plotting data
ex. 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

January 2011

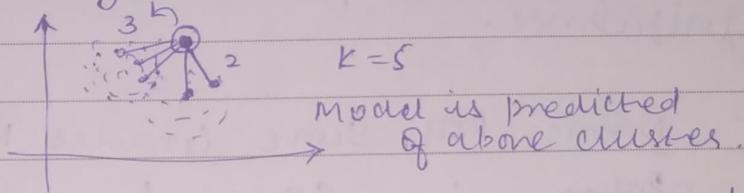
11

Week 2 • 11-354

Instance Based Learning -

Use raw data pts. themselves to figure out outcomes for newer, previously unseen data samples.

Ex. K-nearest.



② Batch Learning.

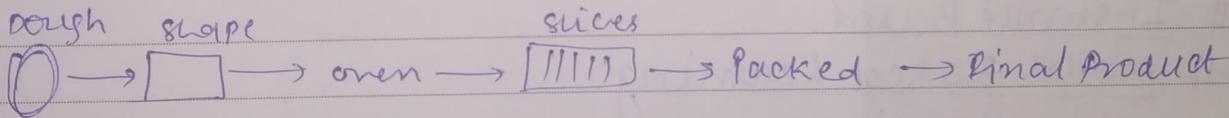
- also called offline learning ~~model~~.

ML method used in end to end ML where the model is trained on avail. data in a single go.

If we get satisfactory result, we deploy it.

drawback - once training is complete, the model stops learning.

Ex. Bread Manufacturer Company. -



If we ask to make a big [bread]. The m/c is not trained on big bread. To make a big bread we need to retrain it.
retraining doesn't occur in Batch learning.

③ Online Learning.

Training data is fed in multiple incremental batches to algorithm.

Ex. Stock Market Forecasting., Device Failures.

Data Batches are known as mini-Batches in ML.

Keeps on learning over a period of time based on new data samples which are sent to it for prediction.

12

Wednesday

January 2011

Week 2 • 12-353

	January	February	March	April	May	June	July	August	September	October	November	December
Wk	Mo	Tu	We	Th	Fri	Sa	Su					
52/5	31					1	2					
1	3	4	5	6	7	8	9					
2	10	11	12	13	14	15	16					
3	17	18	19	20	21	22	23					
4	24	25	26	27	28	29	30					

All methods works on principle of 'Garbage In - Garbage Out'
 If you supply bad data samples to well-trained model, it can start learning sets & patterns that have no real significance.

Ex.	Name	Roll No.	Score	Grades	Research Score	Grant
2/3 Yes	Rohan	1	90	A	95	Yes
	Rohan	2	71	B	42	No
	Rohan	3	91	A	99	Yes

Test data
Irrelevant cols.

Rohan 25 29 D 40

Yes

(2 times 'Rohan' is given grant.)

so Name should not be factor for determining.

It should predict

NO but as 'Name' col is not removed it predict on basis of name

Instance Based

Model-Based Learning:

- Iterative Process

- While training each of I/Ps are expecting some parameters those are called Hyperparameters

The way of learning is decided on these hyperparam.
 (I/P data is used to extract features & models are built based on various model parameters.)

Process

CRISP-DM, Model

Cross-industry standard Process for Data mining

(tried, tested & robust industry standard process model followed for datamining & analytics project)

February 2011						
Mo	Tu	We	Th	Fr	Sa	Su
5	1	2	3	4	5	6
6	7	8	9	10	11	12
7	14	15	16	17	18	19
8	21	22	23	24	25	26
9	28					

Thursday

January 2011

13

Week 2 • 13-352

(what is the objective client want to achieve)

Business Understanding

①



② Data Understanding



③ Data Preparation



④ modeling



⑤ Evaluation



⑥ Deployment



DATA

① Set a goal (what milestone you want to achieve, what timeline, success criteria, assumption..etc)

② collect data, describe data, perform EDA, Quality of data. (Bad or insuff data)

* ③ Data Wrangling, (70-80% time)

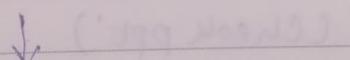
④ Build a ML model.

⑤ If model is running fine - deploy it, if not doing well, check another model & evaluate.

ML pipelines

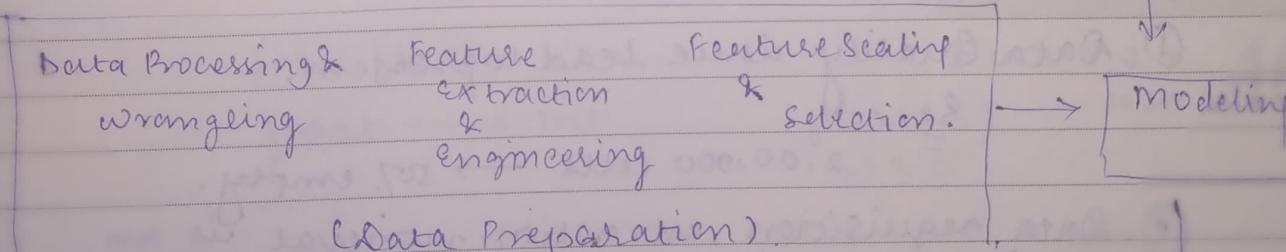
.csv, .db, .json, .xml

Data sets → Data Retrieval

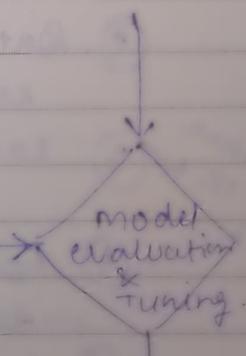


ML

algo



Re-iterate till satisfactory
model performance



Deployment & Monitoring

14

Friday

January 2011

January							2011
Wk	Mo	Tu	We	Th	Fr	Sa	Su
52/5	31					1	2
1	3	4	5	6	7	8	9
2	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23
4	24	25	26	27	28	29	30

Week 2 • 14-351

These are feature sizes

we made a new icon that help in predicting.
(this is feature engineering)

Ex. Name Pic in Total Good/Bad.
 no need now.

Feature Scaling.

#	Ex	price	Normalized-price	#
scale everything	Range 3	1312.22	0.217750	3
Proportionate down.	(same range).	1-2 { 706.62 760.75 }	0.116814 0.125835	lies b/w 1 0 to 1 2
10	4	2445.60	0.406652	4
11		:	:	
12				

1 # Supervised ML pipeline:

Check on

ppt.

flowchart

3 # Unsupervised ML pipeline:

5 # challenges in ML. (Check ppt')

① Data Quality issue lead to problem.

Ex.

2,00,000 cells → 50% empty.

② Data acquisition, extraction, retrieval is an extremely tedious & time consuming process.

Model save
at least 1K rows.

③ Lack of good quality & sufficient training data in many scenarios

④ Formulating Business problem clearly with well defined objectives & goals.

⑤ Feature extraction & engineering → how, why, when.

⑥ Avoid overfitting & underfitting.

⑦, ⑧ ...

	February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa	Su
5		1	2	3	4	5	6
6	7	8	9	10	11	12	13
7	14	15	16	17	18	19	20
8	21	22	23	24	25	26	27
9	28						

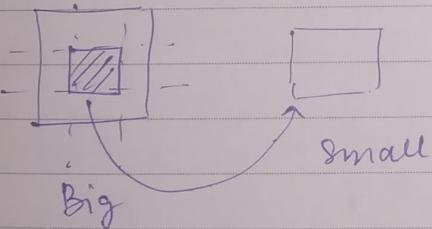
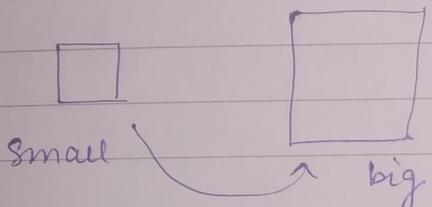
Saturday

January 2011

15

Week 2 • 15-350

Type casting.



int < float
float → int

Sunday
16-349
float(3)
>> 3.0

int ('2020.1245') >> 2020
float ('2020.1245') >> 2020.1245
int ('C') >> error.

and
or
not } logical operators

if 1st cond is 'F' doesn't check 2nd one.

17

Monday

January 2011

Week 3 • 17-348

January 2011						
Wk	Mo	Tu	We	Th	Fr	Sa
52/5	31				1	2
1	3	4	5	6	7	8
2	10	11	12	13	14	15
3	17	18	19	20	21	22
4	24	25	26	27	28	29
					30	

February						
Wk	Mo	Tu	We	Th	Fr	Sa
5					1	2
6					7	8
7					14	15
8					21	22
9					28	

- Bitwise - &, |, ^, ~, <<, >>

'&' is not related to 'and'.

$$\begin{aligned}
 & \text{Ex: } \\
 & \quad x = 2 \\
 & \quad (left shift) \quad x << 1 \quad \# 4 \\
 & \quad \leftarrow 0010 = 2 \text{ (binary)} \\
 & \quad 0100 = 4 \text{ (binary)}
 \end{aligned}$$

- Membership - in, notin

$$\begin{aligned}
 & x >> 1 \quad \left\{ \begin{array}{l} 0010 = 2 \\ 0001 = 1 \end{array} \right. \\
 & (\text{right shift})
 \end{aligned}$$

- Identity - is, is not

$$x = 4$$

$$y = 4$$

$$x \text{ is } y \quad \# \text{ True}$$

$$x \text{ is not } y \quad \# \text{ False.}$$

1 # Conditional statements.

- if else

- elif

+

5 # Nested ifelse Conditions.

① if condn:

≡
elif condn:

≡
elif condn:

≡
else:

≡

② if condn:

≡
elif condn:

≡
if condn:

≡
else:

≡
elif condn:

≡

≡
else:

≡

③ if condn:
if condn:

≡
else:

≡
elif condn:

≡
else:

≡
elif condn:

≡

≡
else:

≡

	February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa	Su
5		1	2	3	4	5	6
6		7	8	9	10	11	12
7		14	15	16	17	18	19
8		21	22	23	24	25	26
9		28					

\ → escape sequence

Tuesday

January

2011

18

Week 3 • 18-347

Looping statements.

- for loop
- while loop

↳ ex

= i=10

while i>=0:

 print(i, end=" ")

 i=i-2

OP: 10 8 6 4 2 0

Nested for.

for i in range(5):

 for j in range(i):

 print("r", end=" ")

 print("\n")

*

**

continue → skips the current iteration.

break → used to terminate loop (for & while).

(if-else)

are not
loop.

Ex.

= n = [10, 20, 30, 40, 50, 60]

for i in n:

 if i == 30:

 break / continue

(Break)

if: 10
20

else:

 print(i)

if there is continue

10
20
40
50
60

19

Wednesday

January 2011

Week 3 • 19-346

S[:: -1]

S[-6:-1]

January

2011

Wk	Mo	Tu	We	Th	Fr	Sa	Su
52/3	31					1	2
1	3	4	5	6	7	8	9
2	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23
4	24	25	26	27	28	29	30

for calling module fn: →
use '•' funsyntax: module.function,Functions.

sqrt(64) # error bcz math fn is not imported.

import math

math.sqrt(64) # 8.0

round
to upper no ↴ math.sin(90) # 0.8939
9 ↴ math.ceil(2.011) # 3
no ↴ math.floor(2.011) # 2

no. # help('math') → you will see all fn of math.

↳ import math as m

1 use alias name (Pet name) : to reduce time & complexity.

m.exp(3.0) # 20.08

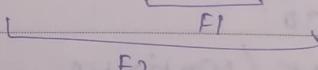
m.log(3) # 1.09

m.pow(2,3) # 8

6 # function composition.

function inside function.

(short) ex print(type('riya'))



user defined function.

Use 'def' keyword.

ex def avg(n1,n2):

sum = n1+n2

print(sum/2)

avg(3,4)

Variables are local in nature.

	February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa	Su
5		1	2	3	4	5	6
6	7	8	9	10	11	12	13
7	14	15	16	17	18	19	20
8	21	22	23	24	25	26	27
9	28						

Thursday

January 2011

20

String functions.

Week 3 • 20-345

s1 = ' Riya '

s2 = ' Dhama '

s1.swapcase() → swap the cases.

s1.lower()

s1.upper()

s1.isalpha() → check if string consist only alphabets.

s1.replace('i', 'z')

s1.find('o', 5) # -1 if it is not present

LIST:

→ check list functions.

Mutable → content can be changed, flexible ..

→ max(s), min(s) ...

s = [1, 2, 3, 4, 5, 6, 7, 8, 9]

s.pop() # last element will be popped.

(LIFO)

s.pop(1) # index at '1' is removed.

s.insert(2, 999) # insert at index 2

s.append(777) # add at end

s.sort()

s.reverse()

s.clear() # clear list content

del s[1] s.remove(s)
delete 1st index.

if there is no 5 in list → O/P → error.

if there are 2 five in
list then only 1
will remove.

max(s)

min(s)

len(s)

{ same
can be
used
for
tuple.

TUPLE:

this { t = (1,) } # type tuple.

it { t = (1.) } # type float

only { d = [1] } # type list

for
round
brackets.

* Note:
for traversing
use
'for' loop.

21

Friday

January 2011

Week 3 • 21-344

January 2011						
Wk	Mo	Tu	We	Th	Fr	Sa
52/53	31					1
1	3	4	5	6	7	8
2	10	11	12	13	14	15
3	17	18	19	20	21	22
4	24	25	26	27	28	29

Variable length arguments.* → any no of arg.
can be accepted.def var(*args):
 print(args)

output:

var(1) # (1)
var(1, 2, 3) # (1, 2, 3)
var('niya', 'letsupgrade', 19) # ('niya', 'letsupgrade', 19)
var() # ()

9

10

11

12

{ concatenation (+)}

1

{ Replication (*)}

2

3

t1 = (1, 2, 3)

4

t2 = (5, 6, 4, 1)

5

t1 * t2

6

t1 + t2

→ key & value are of user's choice.

DICTIONARY: / Associative Array.

- unordered set of key value pairs
- values → mutable
- keys → immutable

Ex: d1 = {1: 'comp', 2: 'IT', 3: 'ExTe'}

d1 d[1]

d1.clear()

- d1.keys()
- d1.values()

delete complete value & key.

empty dict. now.

February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa Su
5		1	2	3	4	5 6
6		7	8	9	10	11 12 13
7		14	15	16	17	18 19 20
8		21	22	23	24	25 26 27
9						

Saturday

January 2011

22

Week 3 • 22-343

You can't have multiple key of same name. If it happens last value updated to that key.

Ex `ol = {'comp': 3, 'extc': 4, 'comp': 99}`

	comp	extc		
d	3 99	4	.	.

It is already in dict. 'd'. so it updates value.

`print(d)`

{'comp': 99, 'extc': 4}

whole

`dl = len(d)` # count the key-value pair. O/P: 2

`dl = {1: 'comp', 2: 'IT', 3: 'extc'}`

`dl2 = {100: 'chem'}`

`dl.update(dl2)` # dl2 is added in dl.

(works similar with: extend fn in list)

`dl['IT']` } Both of them give the
`dl.get('IT')` } value at 'IT' key.

- dictionary can contain list, tuple, dict.

Sunday

23

`dl = {1: ['AB', 'CD'], 2: ['D', 'E']}`

23-342

`dl2 = {1: ('1', 2), 2: ('1', '2'), 3: ['1', '2']}`

`type(dl2)` # dict

`type(dl2[3])` # list

`dl3 = {1: ('1', 2), 2: ('1', '2'), 100: {44: ('1', 2), 45: ('1', '2')}}`

dict

- list can contain tuple, dict.

- tuple can contain list, dict.

24

NOTE : scalar datatypes like
int, float, char
can be keys
in dict

Monday

January

2011

iterable objects like

list, tuple can't be
keys. # single value.

tuple can be
a key.

(try for
single value
list).

Week 4 • 24-341

File Handling.

2 types of files :

→ Text

→ Binary.

'r' → reading

'w' → writing

'a' → append

'r+' → Both read & writing.

NOTE:

- When you open any file, always close that file so to avoid any other obj. to overwritten on that.

9 → file must
10 f = open ('H.txt', 'r') exist if you
11 f.close() are using
 'r' mode.

12 # Here is an error, bcz this file
1 doesn't exist in that folder.

2 → open this in write mode
3 f = open ('H.txt', 'w') it create file automatically
4 f.write ('Hello Riya\n')
5 f.write ('Whatsup?\n')
6 print (f.name) # print file name.

attribute (NOT a fn).

f.close()

True.

print(f.closed)

It will tell whether file is
closed or not.

'w' → if you used this on already created file → it will
then write other content using f.write() you will
lose previous content of file.

'a' → To avoid losing previous content, just use append
mode. (It means after previous content, add new content also
without deleting previous one).

January 2011						
Wk	Mo	Tu	We	Th	Fr	Sa
5/5	31					1 2
1	3	4	5	6	7	8 9
2	10	11	12	13	14	15 16
3	17	18	19	20	21	22 23
4	24	25	26	27	28	29 30

February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa
5	1	2	3	4	5	6
6	7	8	9	10	11	12
7	14	15	16	17	18	19
8	21	22	23	24	25	26
9	28					

Tuesday

January 2011

25

Week 4 • 25-340

```
f = open('hi.txt', 'a')
f.write("How are you?")
f.close()
```

hi.txt
Hello Riya,
whatsup
How are you?

```
f = open('hi.txt', 'r')
print(f.read())
f.close()
```

→ read fn reads all the content
of a file.

f.read(6) → read only 6 characters.

f.readlines() → read only first line.

If you use another readline, it reads 2nd line.

(Mostly use to read content line by line using for loop.)

f.readlines() → gives opp in a list. (It reads the
entire content & makes a list.)

opp: ['Hello Riya\n', 'whatsup\n', 'How are you?']

f.tell() # Tells where is my cursor (last-post).
index value of cursor.

os.function

os.rename('currentfile', 'newname')

import os

os.rename('hi.txt', 'Riya.txt')

os.mkdir('Pycharm')

Create a directory / folder.

os.getcwd() # Path where directory / NB's are stored.

os.chdir("path of choice") # change path.

26

Wednesday

January 2011

Week 4 • 26-339

	January 2011						
Wk	Mo	Tu	We	Th	Fr	Sa	Su
52/5	31					1	2
1	3	4	5	6	7	8	9
2	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23
4	24	25	26	27	28	29	30

os.listdir('Pycharm') # remove directory.

os.rename('Pycharm', 'Nocharm')



os.listdir()

↳ all files in directory are listed below.

map() function:

- applies the given functn to all iterables & returns a new list.

- syntax:

map(function, iterables).

ex1: def sqr(a):

 return a * a

2 n = map(sqr, [1, 2, 3, 4, 5])

 ↓ ↓
 fn iterables.

5 print(n) # <map object at ...>

6 print(list(n))

 # [1, 4, 9, 16, 25]

↓ ↓ ↓
[1, 2, 3, 4, 5]
axa

axa is applied on every item in list.

→ NO output
it returns map object

all in one line.

print(list(map(sqr, [1, 2, 3, 4, 5])))

↓ ↓ ↓
obj. : :
converted to list.

ex2: def sub(a, b):

 return a - b

print(list(map(sub, [10, 20, 30], [1, 2, 3])))

↓ P: [9, 18, 27]

here can
be
list,
tuple,
set.

February 2011						
Wk	Mo	Tu	We	Th	Fr	Sa Su
5		1	2	3	4	5 6
6	7	8	9	10	11	12 13
7	14	15	16	17	18	19 20
8	21	22	23	24	25	26 27
9	28					

Thursday

27

January 2011

Week 4 ■ 27-338

NOT a name,

→ it is a keyword.

lambda function / throw away fn:

- nameless functions.
- generally provided as I/P to other functn.
- single line functions.

- One time use. (use & throw).
- H/P - O/P to other fn.
- reduce code size.

Syntax: lambda arg : expression

Ex1 $\text{sqr} = \lambda a : a * a$
 $\text{sqr}(3)$ # 9

Ex2 def abc(x):
 $\quad \quad \quad \text{return lambda y : x + y}$
 $t = abc(4)$
 print(t) # obj(4+y) → op: <fn abc'. <locals>...
 print(t(5))
 \downarrow give t → 5
 $t = (4+y)$ # op: 9
 $t = 4+5=9$

Ex3: $z = \lambda x, y : -5*x + 3*y$ # linear eqn
 $z(2,4)$ # 22

Use of lambda fn with map fn:

Ex $\text{list(map(lambda x: x+4, [1,2,3,4]))}$

Ans → [5, 6, 7, 8]
 $[5, 6, 7, 8]$

January						
Wk	Mo	Tu	We	Th	Fr	Sa
52/5	31					1
1	3	4	5	6	7	8
2	10	11	12	13	14	15
3	17	18	19	20	21	22
4	24	25	26	27	28	29
						30

28

Friday

January 2011

Week 4 • 28-337

filter() function.

give iterables with help of another fn passed as an arg.
to test all elements to be True or False.

Syntax:

filter(function, iterables)

[1, 2, 3, 4]

Ex. def Sqr(a):
if a >= 3:
return a

type: filter
x = filter(Sqr, [1, 2, 3, 4])
print(list(x)) # [3, 4]

[3, 4]

Ex. Use lambda inside filter function.

1
2 l = [1, 2, 3, 4, 5]
3 Sqr = list(filter(lambda x: x >= 3, l))
4 print(Sqr). # [3, 4, 5]
5

[1, 2, 3, 4, 5]
xx
x

reduce() function.

applies to other fn to a list that are passed as a parameter to it & finally returns a single value.

Syntax:

reduce(function, iterables)

Ex. import functools

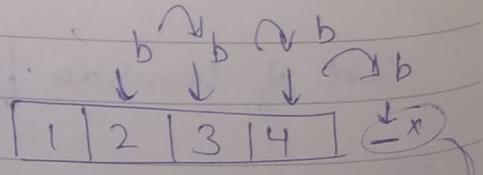
from functools import reduce.

def Sqr(a, b):

return a + b

x = reduce(Sqr, [1, 2, 3, 4])

print(x).



$$\begin{aligned} a &= 1 + 2 = 3 + 3 = 6 + 4 \\ &= 10 \end{aligned}$$

Now it stops.

February 2011						
Wk.	Mo	Tu	We	Th	Fr	Sa Su
5		1	2	3	4	5 6
6		7	8	9	10	11 12 13
7		14	15	16	17	18 19 20
8		21	22	23	24	25 26 27
9		28				

Saturday

January 2011

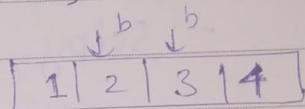
29

Week 4 • 29-336

We need to return $a+b$, b is iterated until the list ends.
at the end it returns a single value. It is similar to
series functions (sum of 50 natural no., ...)

- use reduce with lambda.

Ex. $\text{reduce}(\lambda a,b : a*b, [1,2,3,4])$
24.



$$a=1 \times 2 = 2 \times 3 = 6 \times 4 = 24$$

- filter within map:

Ex. $x = (\text{map}(\lambda n : n+x, \text{filter}(\lambda n : n \geq 3, [1,2,3,4])))$
[6,8]

- map within filter:

Ex. $x = (\text{filter}(\lambda n : n > 6, \text{map}(\lambda n : n+3, [1,2,3,4])))$
[7].

[4,5,6,7]

Sunday

30-335

30

- map & filter within reduce.

Ex. $x = \text{reduce}(\lambda a,y : a+y, \text{map}(\lambda n : n+n, \text{filter}(\lambda n : n \geq 3, [1,2,3,4])))$
[14]

$$6+8=14$$

[3,4]