

PROGRAMMING BASIC(DAY 2)



What we learn today

1. Pseudo Code
2. Why Pseudo Code
3. Variable Type
4. Operator
5. Sequential
6. Condition
7. CALCULATOR program with Pseudo Code
8. Looping
9. Triangle Star Program

Pseudo Code

Pseudo Code , အတုအယောင် code ကတေသ့ program တွေကို စဉ်းစားသည့် အခါမှာ
အသုံးပြုပါတယ်။ Basic က စလွှဲလာသည့် အခါမှာ Pseudo Code, Flow Chart တို့
ကနေ စပီး လွှဲလာဖို့ လိုအပ်ပါတယ်။

Programming က အဓိက အားဖြင့် Logic , စဉ်းစား တွေးခေါ်ပုံက အဓိက ကျပါ
တယ်။ ဘယ် language က ပိုကောင်းတယ် ဆိုတာ ထက် ဘယ်ပြဿနာကို ဘယ်လို့
ဖြေရှင်းရမလဲ ဆိုတာ ပို့ အရေးပါတယ်။

တကယ့်လက်တွေ့ အလုပ်တွေထဲမှာ Backend ကို PHP (Laravel), Front End ကို
React (Javascript) , နဲ့ အချို့ backend system တွေကို Machine Learning (Python) နဲ့
ပေါင်းပြီး ရေးထားတွေ အများကြီး ရှိပါတယ်။

ဒါကြောင့် language ထက် စဉ်းစားတတ် ဖို့ ကို ဦးစားပေးဖို့ လိုပါတယ်။

Pseudo Code

```
If student's grade is greater than or equal to 40
    Print "passed"
else
    Print "failed"
```



Variable

- String
- Integer (-2147483648 through 2147483647)
- Float
- Boolean



Python Variable Bit

- decimal 52
- dict 144
- float 16
- int 14
- list 32
- object 8
- set 112
- str 26
- tuple 24
- unicode 26

1 bits = 1/0 (binary system)

8 bits = 1 bytes ,

1024 bytes = 1 KB ,

1024 KB = 1 MB

1024 MB = 1 GB

1024 GB = 1 TB

THIS IS A SUBHEADLINE

SEQUENCE

READ height of rectangle

READ width of rectangel

COMPUTE area as height times width

PYTHON

```
height = input ("Enter Height of Rectangle")
width = input ("Enter Width of Rectanble")
area = int(height) * int(width)
print("Area is ",area)
```



THIS IS A SUBHEADLINE

CONDITION (IF)

READ height of rectangle

IF height is less than or equal zero

 Print "please enter more than zero"

ELSE IF height is greater than 100

 Print "please enter less than 101"

ELSE

 Print "Height is " + height

PYTHON

```
height = int(input ("Enter Height of Rectangle"))
if height <= 0 :
    print("please enter more than zero")
elif height > 100 :
    print("please enter less than 100")
else:
    print("Height is ",height)
```

```
def __init__(self, settings):
    self.file = None
    self.fingerprints = set()
    self.logdups = True
    self.debug = debug
    self.logger = logging.getLogger(__name__)
    if path:
        self.file = open(os.path.join(settings['job_dir'], 'fingerprint.log'), 'a')
        self.file.seek(0)
        self.fingerprints.update(self._load_file())
@classmethod
def from_settings(cls, settings):
    debug = settings.getbool('general.debug')
    return cls(job_dir=settings['job_dir'], debug=debug)

3
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6098
6099
60100
60101
60102
60103
60104
60105
60106
60107
60108
60109
60109
60110
60111
60112
60113
60114
60115
60116
60117
60118
60119
60119
60120
60121
60122
60123
60124
60125
60126
60127
60128
60129
60129
60130
60131
60132
60133
60134
60135
60136
60137
60138
60138
60139
60140
60141
60142
60143
60144
60145
60146
60147
60148
60148
60149
60150
60151
60152
60153
60154
60155
60156
60157
60158
60159
60159
60160
60161
60162
60163
60164
60165
60166
60167
60168
60169
60169
60170
60171
60172
60173
60174
60175
60176
60177
60178
60179
60179
60180
60181
60182
60183
60184
60185
60186
60187
60188
60188
60189
60190
60191
60192
60193
60194
60195
60196
60197
60197
60198
60199
60199
60200
60201
60202
60203
60204
60205
60206
60207
60208
60209
60209
60210
60211
60212
60213
60214
60215
60216
60217
60218
60218
60219
60220
60221
60222
60223
60224
60225
60226
60227
60228
60228
60229
60230
60231
60232
60233
60234
60235
60236
60237
60238
60238
60239
60240
60241
60242
60243
60244
60245
60246
60247
60248
60248
60249
60250
60251
60252
60253
60254
60255
60256
60257
60258
60258
60259
60260
60261
60262
60263
60264
60265
60266
60267
60268
60268
60269
60270
60271
60272
60273
60274
60275
60276
60277
60278
60278
60279
60280
60281
60282
60283
60284
60285
60286
60287
60288
60288
60289
60290
60291
60292
60293
60294
60295
60296
60297
60297
60298
60299
60299
60300
60301
60302
60303
60304
60305
60306
60307
60308
60308
60309
60310
60311
60312
60313
60314
60315
60316
60317
60318
60318
60319
60320
60321
60322
60323
60324
60325
60326
60327
60328
60328
60329
60330
60331
60332
60333
60334
60335
60336
60337
60338
60338
60339
60340
60341
60342
60343
60344
60345
60346
60347
60348
60348
60349
60350
60351
60352
60353
60354
60355
60356
60357
60358
60358
60359
60360
60361
60362
60363
60364
60365
60366
60367
60368
60368
60369
60370
60371
60372
60373
60374
60375
60376
60377
60378
60378
60379
60380
60381
60382
60383
60384
60385
60386
60387
60388
60388
60389
60390
60391
60392
60393
60394
60395
60396
60397
60397
60398
60399
60399
60400
60401
60402
60403
60404
60405
60406
60407
60408
60408
60409
60410
60411
60412
60413
60414
60415
60416
60417
60418
60418
60419
60420
60421
60422
60423
60424
60425
60426
60427
60428
60428
60429
60430
60431
60432
60433
60434
60435
60436
60437
60438
60438
60439
60440
60441
60442
60443
60444
60445
60446
60447
60448
60448
60449
60450
60451
60452
60453
60454
60455
60456
60457
60458
60458
60459
60460
60461
60462
60463
60464
60465
60466
60467
60468
60468
60469
60470
60471
60472
60473
60474
60475
60476
60477
60478
60478
60479
60480
60481
60482
60483
60484
60485
60486
60487
60488
60488
60489
60490
60491
60492
60493
60494
60495
60496
60497
60497
60498
60499
60499
60500
60501
60502
60503
60504
60505
60506
60507
60508
60508
60509
60510
60511
60512
60513
60514
60515
60516
60517
60518
60518
60519
60520
60521
60522
60523
60524
60525
60526
60527
60528
60528
60529
60530
60531
60532
60533
60534
60535
60536
60537
60538
60538
60539
60540
60541
60542
60543
60544
60545
60546
60547
60548
60548
60549
60550
60551
60552
60553
60554
60555
60556
60557
60558
60558
60559
60560
60561
60562
60563
60564
60565
60566
60567
60568
60568
60569
60570
60571
60572
60573
60574
60575
60576
60577
60578
60578
60579
60580
60581
60582
60583
60584
60585
60586
60587
60588
60588
60589
60590
60591
60592
60593
60594
60595
60596
60597
60597
60598
60599
60599
60600
60601
60602
60603
60604
60605
60606
60607
60608
60608
60609
60610
60611
60612
60613
60614
60615
60616
60617
60618
60618
60619
60620
60621
60622
60623
60624
60625
60626
60627
60628
60628
60629
60630
60631
60632
60633
60634
60635
60636
60637
60638
60638
60639
60640
60641
60642
60643
60644
60645
60646
60647
60648
60648
60649
60650
60651
60652
60653
60654
60655
60656
60657
60658
60658
60659
60660
60661
60662
60663
60664
60665
60666
60667
60668
60668
60669
60670
60671
60672
60673
60674
60675
60676
60677
60678
60678
60679
60680
60681
60682
60683
60684
60685
60686
60687
60688
60688
60689
60690
60691
60692
60693
60694
60695
60696
60697
60697
60698
60699
60699
60700
60701
60702
60703
60704
60705
60706
60707
60708
60708
60709
60710
60711
60712
60713
60714
60715
60716
60717
60718
60718
60719
60720
60721
60722
60723
60724
60725
60726
60727
60728
60728
60729
60730
60731
60732
60733
60734
60735
60736
60737
60738
60738
60739
60740
60741
60742
60743
60744
60745
60746
60747
60748
60748
60749

```

THIS IS A SUBHEADLINE

Operators

- +
- -
- *
- /
- ==
- !=
- >
- >=
- <
- <=
- and
- or

```
def __init__(self, path=None, debug=False):
    self.file = None
    self.fingerprints = set()
    self.logdups = True
    self.debug = debug
    self.logger = logging.getLogger(__name__)
    if path:
        self.file = open(os.path.join(path, 'fingerprint.log'), 'a')
        self.file.seek(0)
        self.fingerprints.update(fingerprint for fingerprint in self.read())
    else:
        self.logdups = False
        self.debug = False
        self.logger.setLevel(logging.INFO)

@classmethod
def from_settings(cls, settings):
    job_dir = settings.get('job_dir', os.getcwd())
    debug = settings.getboolean('debug', False)
    return cls(job_dir=job_dir, debug=debug)

def request_seen(self, request):
    fp = self.request_fingerprint(request)
    if fp in self.fingerprints:
        return True
    self.fingerprints.add(fp)
    if self.file:
        self.file.write(fp + os.linesep)

def request_fingerprint(self, request):
    return request_fingerprint(request)
```

CONDITION (SWITCH)

```
CASE https_status OF
    200 :
        Print "OK"
        BREAK
    404 :
        Print "NOT FOUND"
        BREAK
    500 :
        Print "Internal Server Error"
        BREAK
    OTHERWISE:
        OUTPUT "Unknown Error"
        BREAK
ENDCASE
```

NO CASE IN PYTHON Until 3.10

```
def __init__(self, settings):
    self.file = None
    self.fingerprints = set()
    self.logdups = True
    self.debug = debug
    self.logger = logging.getLogger(__name__)
    if path:
        self.file = open(os.path.join(settings['job_dir'], path), 'a')
        self.file.seek(0)
        self.fingerprints.update(self._load_fingerprints())
@classmethod
def from_settings(cls, settings):
    debug = settings.getbool('debug', False)
    return cls(job_dir=settings['job_dir'], debug=debug)

def request_seen(self, request):
    fp = self.request_fingerprint(request)
    if fp in self.fingerprints:
        return True
    self.fingerprints.add(fp)
    if self.file:
        self.file.write(fp + os.linesep)

def request_fingerprint(self, request):
    return request_fingerprint(request)
```

Calculator Program

```
PRINT "Enter Number 1"  
INPUT number1  
  
PRINT "Enter Operator + , - , * ,  
/"  
INPUT operator  
  
PRINT "Enter Number 2"  
INPUT number2
```

```
IF operator is +  
    PRINT number1 + number2  
ELSE IF operator is -  
    PRINT number1 - number2  
ELSE IF operator is *  
    PRINT number1 * number2  
ELSE IF operator is /  
    PRINT number1 / number2
```

LOOPING

ထပ်ခါထပ်ခါ ဖန်တီးရခြင်း အတွက် lopping ကို အသုံးပြုနိုင်သည်။

ဥပမာ * ကို ၁၀ ခါ ရှိက်မည့် အစား loop ကို ၁၀ ကြိမ် ပတ်ပြီး * ကို ထုတ်နိုင်သည်။

Looping တွင် while loop , for loop ဆုံးပြီး ရှိသည်။

While loop ၏ အဆုံးကို condition ဖြင့် ထိန်းသည်။

FOR loop ၏ အဆုံးကို အရေအတွက် ဖြင့် ထိန်းသည်။

FOR LOOP

```
FOR counter = 1 TO 100 STEP 1 DO  
    OUTPUT counter  
ENDFOR
```

```
for counter in range(1, 101):  
    print(counter)
```

```
FOR counter = 1 TO 100 STEP 3 DO  
    OUTPUT counter  
ENDFOR
```

```
for counter in range(1, 101, 3):  
    print(counter)
```

WHILE LOOP

```
counter is 1  
WHILE counter <= 100 THEN  
    OUTPUT counter  
    counter = counter + 1  
END WHILE
```

```
counter = 1  
while counter <= 100:  
    print(counter)  
    counter = counter + 1
```

TRIANGLE STARS

Program 1

```
*  
**  
***  
****  
*****
```

Program 1

```
FOR row = 1 TO 5 STEP 1 DO  
    FOR col = 1 TO row STEP 1 DO  
        print "*"  
    END FOR  
    print "\n"  
END FOR
```

Program 2

```
*****  
****  
***  
**  
*
```

Python

```
for row in range(1,6):  
    for col in range(1,row + 1):  
        print("*",end="")  
    print("\n",end="")
```

SATURN GOD

THANK YOU