

HACKERS FOR PYTHON

Halit Alptekin (info@halitalptekin.com)

27 Aralık 2014

Süleyman Demirel Üniversitesi Yazılım Klübü

EDUC Bilgisayar mühendisliği

WORK Bilgi güvenliği

IDEA Özgür yazılım ve açık kaynak tutkunu

MEMB TMD, LKD, Octosec

LIFE GO, matematik, kriptoloji

HOBB A sınıfı amatör telsizci, elektronik, robotik

HACKERS

Medya'nın bize gösterdiği **Hacker** tasviri ne kadar doğru?



REAL HACKERS

Bugün bilinen ilk **Hacker** kültürü 1961 yılında MIT'de ortaya çıktı. Tanım itibariyle herhangi bir şeyi amacı dışında kullanan kişilere denilir. Belirledikleri yeni amaca göre farklı isimler alabilirler.



OPEN SOURCE & FREE SOFTWARE

Free Software terimi bir yazılımın "Özgür" olduğunu belirtmek için kullanılır. Bu özgürlük içerisinde çalıştırma, kopyalama, dağıtma, inceleme, değiştirme ve geliştirme gibi kavramlar var.

Sosyal bir harekettir ve çeşitli lisanslar ile sağlanabilir.(GPL, BSD ...)

Open Source terimi bir yazılımın kaynak kodlarının herkes tarafından erişilebilir olduğunu belirtmek için kullanılır. Bir yazılım açık kaynaklı olup, özgür bir yazılım olmayabilir. Ancak her özgür yazılımın kaynak kodları açık olmak durumundadır. Kodları açmak yazılımı özgür bırakmak için yeterli değildir.

CLOSE WINDOWS!

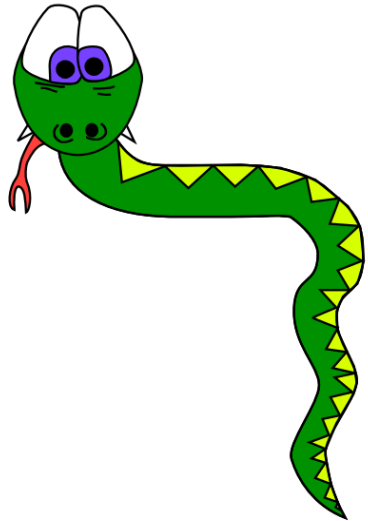


CLOSE WINDOWS,
OPEN DOORS



PYTHON

1990 yılında Guido van Rossum tarafından geliştirilmeye başlanmıştır. İsmi **Monty Python's Flying Circus** isimli gösteriden almıştır. Sadece bir programlama dili olmanın ötesine geçen nadir dillerdendir. Özellikle **Hacker** kültürünün **Perl**'den sonra sahiplenmesiyle ayrıcalıklı bir duruma gelmiştir.



- Okunabilir söz dizimi(syntax)

- Okunabilir söz dizimi(syntax)
- Gelişmiş kütüphane desteği

- Okunabilir söz dizimi(syntax)
- Gelişmiş kütüphane desteği
- Kolay öğrenme

- Okunabilir söz dizimi(syntax)
- Gelişmiş kütüphane desteği
- Kolay öğrenme
- Aktif bir topluluk

- Okunabilir söz dizimi(syntax)
- Gelişmiş kütüphane desteği
- Kolay öğrenme
- Aktif bir topluluk
- Taşınabilir uygulamalar

- Okunabilir söz dizimi(syntax)
- Gelişmiş kütüphane desteği
- Kolay öğrenme
- Aktif bir topluluk
- Taşınabilir uygulamalar
- Çoklu paradigma

- Okunabilir söz dizimi(syntax)
- Gelişmiş kütüphane desteği
- Kolay öğrenme
- Aktif bir topluluk
- Taşınabilir uygulamalar
- Çoklu paradigma
- Hızlı geliştirme

- Okunabilir söz dizimi(syntax)
- Gelişmiş kütüphane desteği
- Çok kolay öğrenme
- Aktif bir topluluk
- Taşınabilir uygulamalar
- Çoklu paradigma
- Hızlı geliştirme

UÇMAK MI İSTİYORSUN?



YOU'RE FLYING!
HOW?



I LEARNED IT LAST
NIGHT! EVERYTHING
IS SO SIMPLE!

HELLO WORLD IS JUST
print "Hello, world!"

I DUNNO...
DYNAMIC TYPING?
WHITESPACE?

COME JOIN US!
PROGRAMMING
IS FUN AGAIN!
IT'S A WHOLE
NEW WORLD
UP HERE!



BUT HOW ARE
YOU FLYING?

I JUST TYPED
import antigravity
THAT'S IT?

... I ALSO SAMPLED
EVERYTHING IN THE
MEDICINE CABINET
FOR COMPARISON.



BUT I THINK THIS
IS THE PYTHON.

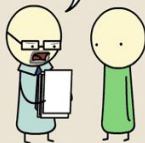
PYTHON

THIS IS PLAGIARISM.
YOU CAN'T JUST "IMPORT" ESSAY."



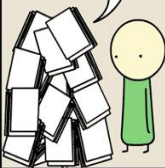
JAVA

I'M TWO PAGES IN AND I STILL
HAVE NO IDEA WHAT YOU'RE SAYING.



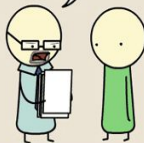
C++

I ASKED FOR ONE COPY.
NOT FOUR HUNDRED.



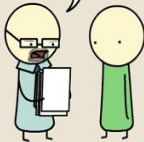
UNIX SHELL

I DON'T HAVE PERMISSION TO
READ THIS.



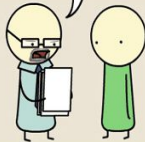
ASSEMBLY

DID YOU REALLY HAVE TO REDEFINE EVERY
WORD IN THE ENGLISH LANGUAGE?



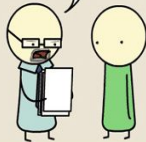
C

THIS IS GREAT, BUT YOU FORGOT TO ADD
A NULL TERMINATOR. NOW I'M JUST READING
GARBAGE.



LATEX

YOUR PAPER MAKES NO GODDAMN SENSE,
BUT IT'S THE MOST BEAUTIFUL THING
I HAVE EVER LAD EYES ON.



HTML

THIS IS A FLOWER POT.



- Web sitesi

NE YAPABİLİRİM?

- Web sitesi
- Oyun

- Web sitesi
- Oyun
- Günlük araçlar

- Web sitesi
- Oyun
- Günlük araçlar
- Mobil uygulama

- Web sitesi
- Oyun
- Günlük araçlar
- Mobil uygulama
- Bilimsel çalışma ...

- Google, Yahoo, NASA

- Google, Yahoo, NASA
- Dropbox, Disqus, Mozilla

- Google, Yahoo, NASA
- Dropbox, Disqus, Mozilla
- Friendfeed, Reddit, Eventbrite

- Google, Yahoo, NASA
- Dropbox, Disqus, Mozilla
- Friendfeed, Reddit, Eventbrite
- Walt Disney, Battlefield 2, Civilization 4

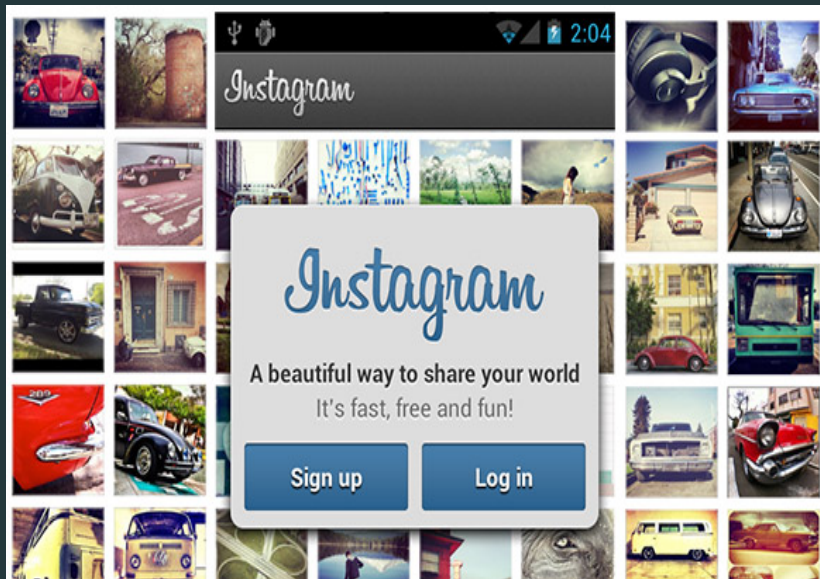
KIM KULLANIYOR?

- Google, Yahoo, NASA
- Dropbox, Disqus, Mozilla
- Friendfeed, Reddit, Eventbrite
- Walt Disney, Battlefield 2, Civilization 4
- Nokia, IBM, CIA ...

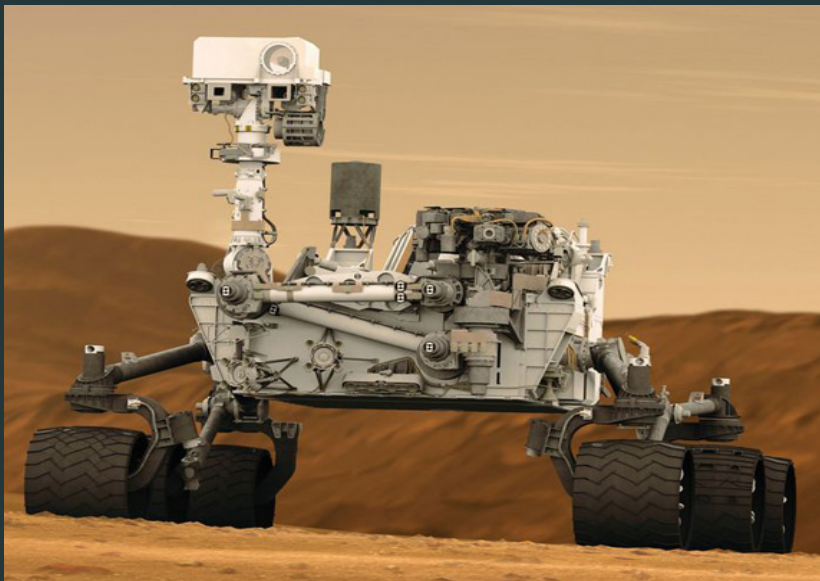
KARAYIP KORSANLARI



INSTAGRAM



YOK ARTIK?



Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

...

There should be one– and preferably only one –obvious way to do it.

...

Namespaces are one honking great idea – let's do more of those!

Python'un kendi yapıları ve veri tiplerini okunabilir, temiz bir şekilde kullanarak oluşturulan ifadelere **Pythonic** denilir.

Herkes **Python** yazabilir ama sadece çok az kişi **Pythonic** kod yazabilir!

Şu anda **Python** ile uğraşmak isteyen birçok kişi hangi sürümü kullanması gerektiği konusunda kafaları karışık. Kafa karışıklığına gerek yok, ikisi de **Python**!

ENVIRONMENT

Vim python.vim, python-mode, jedi ...

Emacs auto-complete, jedi.el ...

Sublime Text anaconda, djaneiro ...

PyCharm Full Python IDE ...

```

5 d./g/index c/box.yml
p - previous diff of selected and current
<cr> - revert to selected state

[10] 3 minutes ago
[9] 3 minutes ago
[8] 3 minutes ago
[7] 6 days ago
[6] 6 days ago
[5] 6 days ago
[4] 6 days ago
[3] 6 days ago
[2] 6 days ago
[1] 6 days ago
[0] Original

Gundo > Undo tree
--- 9 2013-01-25 02:40:12 am
+++ 10 2013-01-25 02:40:14 am
00 -23,12 +23,7 00
- google_analytics/
- google_cse/
- mediagenerator/
- filters/
- generators/
- management/
- templatetags/
- _init_.pyc
- concat.pyc
- cssurl.pyc
- sass.pyc
- yuicompressor.pyc
- _init_.pyc
- _generated_media_names.py
- _generated_media_names.pyc
- migrate.py
- app.yaml
- backup.sh
- blank.html
- cleanhtmlwriter.py
- cleanhtmlwriter.pyc

+ blog/
+ caching/
+ dbindexer/
+ Disqus/
+ django/
+ djangoengine/
+ djangoext/
+ docutils/
+ events/
+ google_analytics/
+ google_cse/
+ mediagenerator/
+ filters/
+ generators/
+ management/
+ templatetags/
+ _init_.pyc
+ base.pyc
+ middleware.pyc
+ settings.pyc
+ utils.pyc
+ minicas/
+ object_permission_backend/
+ permission_backend_norel/
+ plogo/
+ profiles/
+ program/
+ pygments/
+ redirects/
+ replay/
+ templates/
+ admin/
+ change_form.html
+ submit_line.html
+ _init_.py
+ _init_.pyc
+ admin.py
+ admin.pyc
+ models.py
+ models.pyc
+ robots/
+ script/
+ search/
+ simplesocial/
+ staticstuff/
+ templates/
+ urlrouter/
+ _init_.py
+ _init_.pyc
+ _generated_media_names.py
+ _generated_media_names.pyc
+ migrate.py
+ app.yaml
+ backup.sh
+ blank.html
+ cleanhtmlwriter.py
+ cleanhtmlwriter.pyc

+ blog/
+ caching/
+ dbindexer/
+ Disqus/
+ django/
+ djangoengine/
+ djangoext/
+ docutils/
+ events/
+ google_analytics/
+ google_cse/
+ mediagenerator/
+ filters/
+ generators/
+ management/
+ templatetags/
+ _init_.pyc
+ base.pyc
+ middleware.pyc
+ settings.pyc
+ utils.pyc
+ minicas/
+ object_permission_backend/
+ permission_backend_norel/
+ plogo/
+ profiles/
+ program/
+ pygments/
+ redirects/
+ replay/
+ templates/
+ admin/
+ change_form.html
+ submit_line.html
+ _init_.py
+ _init_.pyc
+ admin.py
+ admin.pyc
+ models.py
+ models.pyc
+ robots/
+ script/
+ search/
+ simplesocial/
+ staticstuff/
+ templates/
+ urlrouter/
+ _init_.py
+ _init_.pyc
+ _generated_media_names.py
+ _generated_media_names.pyc
+ migrate.py
+ app.yaml
+ backup.sh
+ blank.html
+ cleanhtmlwriter.py
+ cleanhtmlwriter.pyc

1 # Git branch master
2 # Untracked files:
3 # (use "git add <file>..." to include in what will be committed)
4 #
5 # .dropbox.js.swp
6 nothing added to commit but untracked files present (use "git add" to track)

N | master > dropbox.js/git/index = 100 unix < utf-8 < gitcommit 164 % 1:
131 xhr.onreadystatechange = function(evt) {
132     if (this.readyState == 4 && this.status == 200) {
133         successCallback(evt);
134     }
135 };
136 xhr.overrideMimeType(data.mimeType);
137 xhr.setRequestHeader('Content-Type', data.mimeType);
138 xhr.sendAsBinary(evt.target.result);
139 }
140 reader.readAsBinaryString(data.fileObj);
141 };
142 c._getRoot = function() {
143     return this.session.accessType == 'dropbox' ? 'dropbox' : 'sandbox';
144 };
145 c._getPath = function(params) {
146     if (!params || !params.path) {
147         throw new Error("Must provide a path to get metadata from");
148     }
149     var path = params.path;
150     delete params.path;
151     return path;
152 };
153 c._getFileUrl = function(prefix, params) {
154     var root = this._getRoot();
155     var path = this._getPath(params);
156     return [prefix, root, path].join('/');
157 };
158 c.accountInfo = function(successCallback, errorCallback) {
159     this.api('account/info', {}, successCallback, errorCallback);
160 };
161 c.metadata = function(params, successCallback, errorCallback) {
162     var url = this._getFileUrl('metadata', params);
163     this.api(url, params, successCallback, errorCallback);
164 };
165 c.putFile = function(params, successCallback, errorCallback) {
166     params.override = params.override || false;
167     params.mimeType = params.fileObj.type || 'text/plain';
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
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

justthis.lan : /Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEngine-default.bundle/Contents/Resources/google_appengine/google_appengine...

GoogleAppEngine-default.bundle
├── Contents
│   ├── MacOS
│   └── Resources
│       ├── google_appengine
│       │   ├── demos
│       │   └── google
│       │       ├── appengine
│       │       │   ├── api
│       │       │   ├── base
│       │       │   ├── datastore
│       │       │   ├── ext
│       │       └── admin
│       ├── google_appengine
│       │   ├── api
│       │   ├── base
│       │   ├── datastore
│       │   ├── ext
│       └── admin
└── google_appengine/google_appengine/ext/db

W-0 ...rces/google_appengine/google_appengine/ext/db
├── idjangoforms.py
├── __init__.py
└── __init__.pyc

W-0 ...rces/google_appengine/google_appengine/ext/db
├── Parents
├── Variables
├── validate (self,value)
├── Misc
├── DateTimeProperty : class
├── Parents
├── Variables
├── __init__ (self,verbose_name,auto_now,auto_now_add,now,now_add,now_func,now_func_args,now_func_kwargs,default_value(self),get_value_for_datastore (self,model_instance),now(),validate (self,value))
├── Misc
├── models.py
├── __init__.py
└── __init__.pyc

W-0 History
Variables: auto_now_add

data_type = Blob

class DateTimeProperty(Property):
    """The base class of all of our date/time properties.

    We handle common operations, like converting between time tuples and
    datetime instances.
    """
    def __init__(self, verbose_name=None, auto_now=False, auto_now_add=False,
        **kwargs):
        """Construct a DateTimeProperty

        Args:
            verbose_name: Verbose name is always first parameter.
            auto_now: Date/time property is updated with the current time every time
                it is saved to the datastore. Useful for properties that want to track
                the modification time of an instance.
            auto_now_add: Date/time is set to the when its instance is created.
                Useful for properties that record the creation time of an entity.
        """
        super(DateTimeProperty, self).__init__(verbose_name, **kwargs)
        self.auto_now = auto_now
        self.auto_now_add = auto_now_add

    def validate(self, value):
        """Validate datetime.

        Returns:
            A valid value.

        Raises:
            ValueError if property is not instance of 'datetime'.
        """
        value = super(DateTimeProperty, self).validate(value)
        if value and not isinstance(value, self.data_type):
            raise ValueError("Property %s must be a %s" %
                (self.name, self.data_type.__name__))
        return value

    def default_value(self):
        """Default value for datetime.

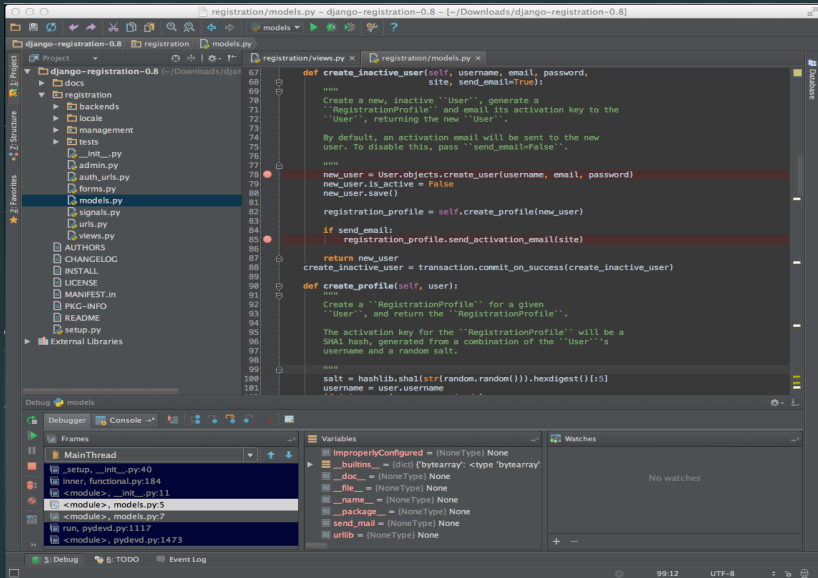
        Returns:
            value of now() as appropriate to the date-time instance if auto_now
            or auto_now_add is set, else user configured default value implementation.
        """
        if self.auto_now or self.auto_now_add:
            return now()
        return self._default_value

W-0 ...rces/google_appengine/google_appengine/ext/db
├── Parents
├── Variables
├── validate (self,value)
├── Misc
├── DateTimeProperty : class
├── Parents
├── Variables
├── __init__ (self,verbose_name,auto_now,auto_now_add,now,now_add,now_func,now_func_args,now_func_kwargs,default_value(self),get_value_for_datastore (self,model_instance),now(),validate (self,value))
├── Misc
├── models.py
├── __init__.py
└── __init__.pyc

W-0 History
Variables: auto_now_add

```

PYCHARM



Python'un kendi paket yöneticisi:

```
apt-get install python-pip  
yum install python-pip  
pacman -S python2-pip (python-pip)
```

Temel komutları:

```
pip install <paket_adi>  
pip search <paket_adi>  
pip uninstall <paket_adi>  
pip install -r requirements.txt
```

Başındaki **i** elmadan gelmiyor! İnteraktif'in kısaltması olduğu için orada duruyor.

```
pip install ipython
```

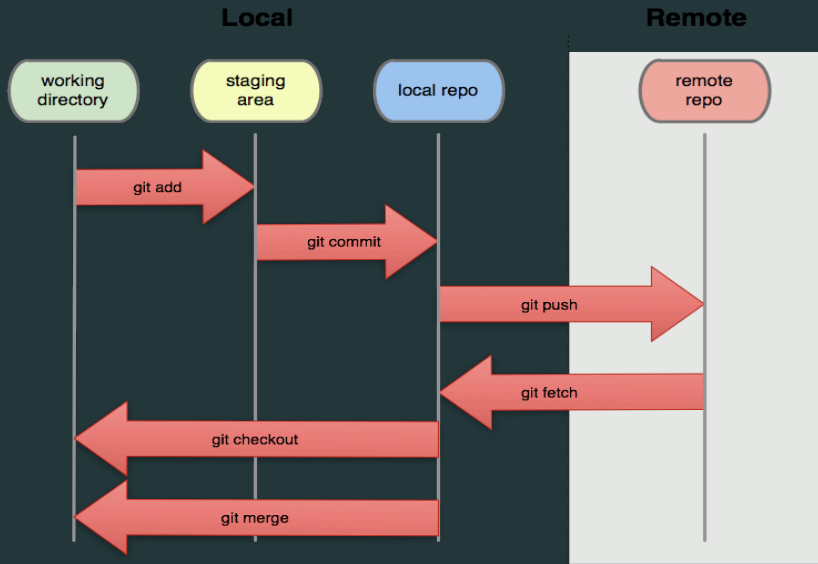
Biz yazacağız o tamamlayacak, biz isteyeceğiz o yardım edecek, biz yap diyeceğiz o yapacak. Ee o zaman bu IDE'ler ne olacak?

Herşeyin zamanla öğrenilecek kullanılması ve kullanılmaması gereken bir yeri var. Acele yok Rocky! Parkur uzun...

Sürüm kontrol ve kaynak kod yönetim yazılımıdır. Biliyorum çok fazla kodun yok ama elbet bir gün olacak. Şimdiden öğrenmen lazım...

```
apt-get install git  
yum install git  
pacman -S git
```

OĞLUM BAK GIT!



Temel komutlar:

```
git init
git add <dosya_adi>
git status
git log
git checkout -- <dosya_adi>
git branch <dal>
git checkout <dal>
git branch -D <dal>
git pull <repo>
git push <repo>
```

Git depolarımızı ücretsiz olarak barındırma hizmeti sağlıyor. Bilinen birçok büyük proje aktif olarak **Github** depolarında geliştiriliyor.

```
git config --global user.name "isim"  
git config --global user.email "e-posta"
```

HTTPS kullanacaksak depoları çekerken **kullanıcı adı** ve **şifre** girmemizi ister. **SSH** kullanacaksak **SSH** keyi oluşturup **Github** içerisinde tanıtmalıyız.

GITHUB ÇOK GÜZEL GELSENIZE!



twbs / bootstrap

Watch

4,465

Star

75,817

Fork

28,847

The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web. <http://getbootstrap.com>

Code



Facebook



We work hard to contribute our work back to the web, mobile, big data, & infrastructure communities. NB: members must have two-factor auth.

Menlo Park, California

<https://code.facebook.com/projects/>

Filters

Find a repository...

mysql-5.6

C++ ★ 204 83

Facebook's branch of the Oracle MySQL v5.6 database

Updated 12 hours ago

rocksdb

C++ ★ 2,909 541

A library that provides an embeddable, persistent key-value store for fast storage.

Updated 12 hours ago

Python bağımlılıkları için izole bir ortam oluşturuyor. Yüklediğiniz paketler sistemin genelini etkilemiyor. Paketlerin farklı sürümleri ile çalışma imkanı sağlıyor.

```
pip install virtualenv
```

Temel komutlar:

```
cd <klasor_adi>
virtualenv <sanal_ortam_adi>
virtualenv -p /usr/bin/python2.7 <sanal_ortam_adi>
source <sanal_ortam_adi>/bin/activate
deactivate
```

Birleşmenin tam zamanı!

```
pip freeze > requirements.txt
pip install -r requirements.txt
```

FUNDAMENTALS

HAZIR MIYIZ?



DAHASINI MI ISTIYORSUNUZ?



Python2:

```
print "nbr dunya"  
print "sn msglsn glba.s.s"
```

Python3:

```
print("nbr dunya")  
print("sn msglsn glba.s.s")
```

Oluşturduğunuz her `py` uzantılı dosyalar birer Python modülüdür. Ancak doğrudan çalışmasını da sağlayabilirsiniz. Bir önceki slaytta yazan satırları `hello.py` olarak kaydedebilirsiniz.

```
python hello.py
```

*nix ortamında çalıştırılabilir dosya yapmak için, dosyanın en başına aşağıdaki satır eklenir.

```
#!/usr/bin/env python
```

Son olarak:

```
chmod +x hello  
./hello
```

Python'da her şey bir nesnedir. Bu nesneler kimliklerini belirten bir `id` ve değerini tutan `value` sahibidir.

```
sdu = 3.14159265538979323846  
id(sdu)
```

Sahip oldukları değerler `değiştirilebilir(mutable)` ve `değiştirilemez(immutable)` olabilir.

Mutable değerler Üzerinde değişiklik yaptığınızda **id** değeri değişmez. Ancak **Immutable** değerler üzerindeki değişiklikler **id** değerini değiştirir.

```
sdu = 3.14159265538979323846
```

```
id(sdu)
```

```
sdu += 1.0
```

```
id(sdu)
```

```
sepet = ["elma", "armut", "kiraz"]
```

```
id(sepet)
```

```
sepet.append('karpuz')
```

```
id(sepet)
```

```
en_buyuk = "besiktas"
```

```
id(en_buyuk)
```

```
en_buyuk += "tir"
```

```
id(en_buyuk)
```

Bir Çin atasözü der ki, güzel kod güzel değişken isminden belli olur. Ayrıca bir Türk atasözü der ki, bana değişken adını söyle sana nasıl bir kod olduğunu söyleyim.

```
ahmet = 1
osman = 1.0
altan = "ohooneleryazilirburaya"
ceyda_osman_kemal = []
ayse_ezgi_riza = {}
```

2 + 2



Matematiksel işlemler için diğer dillerden bilinen $+$, $-$, $/$, $*$ operatörleri aynen geçerlidir. Farklı olarak üs alma işlemi için $**$ ve $\%$ vardır.

```
ahmet = 5 + 6  
ahmet += 2  
ahmet = 3 * 5  
ahmet = 2**5  
ahmet = ahmet % 5  
ahmet %= 5  
ahmet *= 10
```

Tam sayı bölme işlemi diğer dillerde olduğu gibi karıştırılabilir.

```
ahmet = 5 / 2  
ahmet = 5 / 2.0  
ahmet = 5.0 / 2
```

Tam sayı olarak bellekte tutulan en büyük değerin üstüne çıkarsanız, artık o sayıyı **long** tipinde saklamaya başlarsınız.

```
import sys  
ahmet = sys.maxint  
type(ahmet)  
ahmet += 1  
type(ahmet)
```

Metinler karakter dizisi olarak tutulur. Python'da bu karakter dizileri `str` tipindedir. Ayrıca stringler `immutable` bir veri tipidir.

```
ahmet = "iste bu bir string"  
ahmet += "tir."
```

Python'da stringleri tek tırnak, çift tırnak veya üç tırnak ile yazabilirsiniz. Kullanım sebepleri çeşitli sorunlar ile karşılaşınca daha rahat anlaşılır.

```
ahmet = "iste bu bir string"  
ahmet = 'ama bu da bir string'  
ahmet = """vallaha bu da string"""
```

Stringler üzerinde çeşitli formatlama işlemleri yapılabilir.

```
ahmet = "%s %s" %('bu bir', 'string')  
ahmet = "{0} {1}".format('bu bir', 'string')
```

Veri tiplerinin hangi metodlara sahip olduğunu ezberlemenize gerek yok. Python içerisinde tanımlı gelen `help`, `dir` fonksiyonları size yardımcı olacaktır.

```
ahmet = "bu string"  
melda = 3.14159165538979323846  
dir(ahmet)  
dir(melda)  
help(ahmet.upper)  
help(ahmet.title)  
help(melda.hex)
```


`dir` komutunu kullandığınız zaman iki tane alt çizgi ile başlayan(double underscore, dunder) metodlar olduğunu da göreceksiniz. Onlar normalde doğrudan kullanmayacağınız ancak başka bir nesne ile etkileşime girdiğinde kullanacağınız metodlardır.

```
ahmet = 2  
melda = 3  
ceyda = ahmet.__add__(melda)
```

Python'da yorumlar `#` ile başlar. C'de olduğu gibi bir kerede çok satır içeren yorum satırını doğrudan oluşturmakta kullanılmaz. Ancak o amaç için yapılmış başka bir yorum türü vardır.

Python'da ilerlemenin en temel şartı sahip olduğu veri tiplerine hakim olmaktır.

NoneType:

```
ahmet = None
```

Boolean:

```
ahmet = True  
melda = False
```

Python içerisinde veri tiplerini kıyaslamak için `>`, `<`, `>=`, `<=`, `==`, `!=` operatörlerini kullanabilirsiniz.

```
"ahmet" > "mehmet"  
"ahmet" == "ahmet"  
546 != 457  
x = 3.14159265538979323846  
2 < x < 5  
isinstance(x, float)  
isinstance("ahmet", str)
```

Mantıksal işlemler için `and` ve `or` kullanılır.

Farklı tür verileri birarada tutmak için Python `liste(list)`, `sözlük(dict)`, `küme(set)` ve `demet(tuple)` veri tiplerine sahiptir.

```
meyveler = ["elma", "armut"]  
elma = list("elma")
```

```
meyveler = {"elma": "kirmizi", "armut": "sari"}  
meyveler = dict()
```

```
meyveler = {"elma", "armut", "kiraz", "elma"}  
meyveler = set()
```

```
meyveler = ("elma", )  
meyveler = ("elma", "armut")  
meyveler = tuple()
```

Python sözlükleri veri yapısı olarak **hashmap** veya **ilişkili dizi** olarak da bilinir.

```
meyveler = {"elma": "kirmizi", "armut": "sari"}  
meyveler["elma"] = "yesil"
```

in ifadesi nesnenin **__contains__** metodunu kullanmaktadır.

```
print "armut" in meyveler  
print "kiraz" in meyveler
```

get metodu ile erişim yapılabilir ayrıca **del** metodu ile de silinebilir.

```
meyveler["kiraz"]  
meyveler.get("kiraz", "yok ki")  
del meyveler["kiraz"]
```

```
ahmet = "hihahoo"  
meyveler = ["elma", "kiraz", "armut"]  
turgut = "elma,kiraz,armut"  
print ahmet[0]  
print ahmet[-1]  
print meyveler[0:1]  
print meyveler[:1]  
print meyveler[:-1]  
print meyveler[::-1]  
print ahmet[::-1]  
print len(ahmet)  
print turgut.split(',')  
print len(turgut.split(',')) == len(meyveler)  
ahmet.sort()  
print ahmet.sorted()  
elma, kiraz, armut = meyveler
```

```
x = int(raw_input("gir bakim: "))  
if x < 0:  
    print "negatif"  
elif x < 10:  
    print "0 ile 10 arasinda"  
else:  
    print "ne bicim sayi bu?"
```


For:

```
meyveler = ["elma", "armut", "kiraz", "muz"]  
for meyve in meyveler:  
    print meyve, len(meyve)
```

While:

```
meyveler = ["elma", "armut", "kiraz", "muz"]  
i = 0  
while i < len(meyveler):  
    print meyveler[i], len(meyveler[i])
```

Diğer dillerde olan `break` ve `continue` deyimleri Python içerisinde de vardır.

```
sayilar = [2, 3, 1, 5, -6, 7, -8, -2, -1]
for sayi in sayilar:
    if sayi < 0:
        break

for sayi in sayilar:
    if sayi < 0:
        continue

for sayi in range(100):
    pass

for mk in meyveler.keys():
    ...
for mv in meyveler.values():
    ...
for mi in meyveler.items():
    ...
```

Python'da fonksiyonları `def` deyimi ile birlikte tanımlarız. Dikkat edilmesi gereken en önemli nokta 4 boşluk bırakılıp bırakılmadığıdır. Ayrıca fonksiyonların ilk satırı `docstring` olarak isimlendirilir. Buraya yazdıklarınız `help` metodu ile okunabilir.

```
def ekranci():  
    """ ekrana acayip seyler basar """  
    print "acayip bisi"  
  
def eklemeci(a, b, c, d, e=23):  
    """ verdigin her sey ekler birbirine """  
    return a + b + c + d + e  
  
ekranci()  
eklemeci(1, 2, 3)  
eklemeci(1, 2, 3, 4)
```

```
l = "elma armut kiraz karpuz muz"
t_m = [m for m in l.split() if len(m) > 3]
t_s = [[m.upper(), m.lower(), len(m)] \
        for m in l.split()]
t_l = [s for s in range(0, 100) if s % 2]
t_k = {s for s in range(0, 100) if s % 2}
i_s = {s:s+1 for s in range(0, 100) if s % 2}
```

```
tek_mi = lambda x: x % 2 == 1
arttirsana = lambda x: x + 1
sayilar = ["1", "532", "45", "12356"]
sayilar = map(int, sayilar)
sayilar = map(lambda x: x + 1, sayilar)
sayilar_b = map(lambda x: x % 2 == 1, sayilar)
sayilar_t = filter(lambda x: x % 2 == 1, sayilar)
toplama = reduce(lambda x, y: x + y, range(10))
```

Aslında her Python kodu içeren dosya bir modüldür. Modüller içerisinde fonksiyon, sınıf tanımlamaları ile çalıştırılabilir satırlar yer alır. Her modül sadece **import** edildiğinde çalıştırılır. Modüller içerisinde **private** sembol tabloları da bulunur.

```
import modul
modul.fonksiyon()
modul.__name__
fonksiyon = modul.fonksiyon()
fonksiyon()
from modul import fonksiyon1, fonksiyon2
fonksiyon1()
fonksiyon2()
from modul import *
```

```
class Hayvan(object):  
    """ Hayvan sinifi """  
    def __init__(self, isim):  
        """ Yapici metodumuz """  
        self.isim = isim  
    def konus(self):  
        """ Konusturan metodumuz """  
        print "Ben hayvanim: ", self.isim  
  
sari_kiz = Hayvan("sari kiz")  
prens = Hayvan("prens")  
  
sari_kiz.konus()  
prens.konus()
```

```
class Okuz(Hayvan):
    def konus(self):
        """ Konusturan metodumuz """
        print "Ben okuzum: ", self.isim

sari_kiz = Okuz("sari kiz")
sari_kiz.konus()
```



```
dosya = open("dosya.txt", "r")  
for satir in dosya:  
    print satir  
dosya.close()
```

```
dosya = open("dosya.txt", "w")  
dosya.write("yaz beni")  
dosya.close()
```

```
yazdigin = raw_input("Yaz bakim: ")
```

```
import sys  
print sys.argv
```

```
try:
    lay_lay_lom = 123456 / 0
except ZeroDivisionError:
    print "oha 0'a bolmeye calistin oha!"
else:
    print "yok ya 0'a bolmedi"

...
def cok_iyi_fonksiyon(a, b):
    if b != 0:
        return a/b
    else:
        raise CokIlginchHata
...
```

PYTHONIC

```
import this
```

```
temp = ahmet  
ahmet = mehmet  
mehmet = temp  
  
ahmet, mehmet = mehmet, ahmet
```

```
sonuc = None
if 'dizin' in ayarlar:
    sonuc = ayarlar['dizin']
else:
    sonuc = on_tanimli

sonuc = ayarlar.get('dizin', on_tanimli)
```

```
sonuc_listesi = ["True", "False", "Yok"]  
sonuc_stringi = ""  
for sonuc in sonuc_listesi:  
    sonuc_stringi += sonuc  
  
sonuc_listesi = ["True", "False", "Yok"]  
sonuc_stringi = "".join(sonuc_listesi)
```

```
dosya = open("dosya.txt", "r")
for satir in dosya:
    uber_sonik_islem(satir)
dosya.close()

with open("dosya.txt", "r") as dosya:
    for satir in dosya:
        uber_sonik_islem(satir)
```



```
if i == "S" or i == "D" or i == "U":  
    uber_sonik_islem()  
  
if i in ("S", "D", "U"):  
    uber_sonik_islem()
```

```
ss = range(1, 100)
l = []
for s in ss:
    if asal_mi(s):
        l.append(s+5)

ss = range(1, 100)
l = [s+5 for s in ss if asal_mi(ss)]
```

```
i = 0
for eleman in listem:
    print i, eleman
    i += 1

for i, eleman in enumerate(listem):
    print i, eleman
```

```
def liste_olusturucu(arg):  
    listem = []  
    for i in uber_sonik_islem(arg):  
        listem.append(i)  
    return listem
```

```
def liste_olusturucu(arg):  
    for i in uber_sonik_islem(arg):  
        yield i
```

```
sayilar = [1, 5, 2, 4, 3]
for i in range(len(sayilar)):
    sayilar[i] += 3

map(lambda x: x+3, sayilar)
```

```
toplam = 10
for i in (1, 2, 3):
    toplam = toplam ** i

reduce(lambda x, y: x**y, [1, 2, 3], 10)
```

```
sayilar = [5, 6, 3, 4, 7, 9, 1]
yeni_sayilar = []
for i in range(len(sayilar)):
    if sayilar[i] % 2 == 0:
        yeni_sayilar.append(sayilar[i])

filter(lambda x: x%2 == 0, sayilar)
```

```
map(lambda x: x+3, listem)  
[i+3 for i in listem]
```

```
filter(lambda x: x%2 == 0, listem)  
[i for i in listem if i % 2 == 0]
```


LIBRARIES

```
import itertools

itertools.count(10, 1)
itertools.count(10, 2)

itertools.cycle("ABCD")

itertools.repeat(10, 3)

itertools.chain(l1, l2, ...)

itertools.compress("ABCDE", [1, 0, 0, 1])

itertools.product("ABCDE", repeat=2)
itertools.product("ABCDE", repeat=3)
```

```
import collections

meyveler = collections.deque()
meyveler.append("elma")
meyveler.append("armut")

sayac = collections.Counter(buyuk_metin)
sayac.most_common()

ilginc_sozluk = collections.defaultdict(int)
ilginc_sozluk['aboo'] += 50
```

```
meyveler = collections.deque()  
%timeit meyveler.appendleft("elma")
```

```
meyveler = []  
%timeit meyveler.insert("elma", 0)
```

```
pip install requests
```

```
import requests  
istek = requests.get("http://blabla.com")  
istek.text  
istek.headers  
istek.status_code  
istek.ok
```

```
import re
import requests
from collections import Counter
metin = requests.get("http://blabla.com").text
kelimeler = re.findall('\w+', metin.lower())
sayac = Counter(kelimeler).most_common(10)
```

```
import collections
meyveler = \
    [('elma', 1), ('kiraz', 2), \
     ('elma', 3), ('armut', 1), ('karpuz', 4)]
duzgun_liste = collections.defaultdict(list)
for anahtar, deger in meyveler:
    duzgun_liste[anahtar].append(deger)
```

```
import functools
ikili = int("1010")
onlu = int("1010", base=2)
ikili_yap = functools.partial(int, base=2)
ikili_yap("1010")
```



```
import pdb  
pdb.set_trace()
```

h help

s step into

n next

c continue

w where

l list

```
[genel_ayarlar]
site_adi = http://www.google.com
kullanici_adi = root
parola = g00gl3

okuyucu = SafeConfigParser()
okuyucu.read('ayarlar.conf')

print okuyucu.get('genel_ayarlar', 'site_adi')
```

```
import logging
LOG_DOSYASI = "logcu.out"
logging.basicConfig(filename=LOG_DOSYASI,
                    level=logging.DEBUG)
logging.debug("neler oldu neler")
```

```
from bottle import route, run, template

@route('/selam/<isim>')
def index(isim):
    cevap = '<b>Merhaba {{isim}}</b>!'
    return template(cevap, isim=isim)

run(host='localhost', port=8080)
```

Eğitim slaytlarına ve kaynak kodlara aşağıdaki linkten erişebilirsiniz

github.com/halitalptekin/sdu-workshop

SORULAR?