

# Assignment PPL

## API Documentation

April 10, 2017

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Module boy</b>	<b>2</b>
1.1 Variables . . . . .	2
1.2 Class Boy . . . . .	2
1.2.1 Methods . . . . .	2
1.2.2 Instance Variables . . . . .	2
<b>2 Module choosy</b>	<b>3</b>
2.1 Variables . . . . .	3
2.2 Class choosy . . . . .	3
2.2.1 Methods . . . . .	3
2.2.2 Instance Variables . . . . .	3
<b>3 Module couple</b>	<b>4</b>
3.1 Variables . . . . .	4
3.2 Class Couple . . . . .	4
3.2.1 Methods . . . . .	4
3.2.2 Instance Variables . . . . .	4
<b>4 Module desperate</b>	<b>5</b>
4.1 Variables . . . . .	5
4.2 Class desperate . . . . .	5
4.2.1 Methods . . . . .	5
4.2.2 Instance Variables . . . . .	5
<b>5 Module essential</b>	<b>6</b>
5.1 Variables . . . . .	6
5.2 Class Essential . . . . .	6
5.2.1 Methods . . . . .	6
<b>6 Module geek</b>	<b>7</b>
6.1 Variables . . . . .	7
6.2 Class Geek . . . . .	7
6.2.1 Methods . . . . .	7
6.2.2 Instance Variables . . . . .	7
<b>7 Module generous</b>	<b>8</b>

7.1	Variables . . . . .	8
7.2	Class Generous . . . . .	8
7.2.1	Methods . . . . .	8
7.2.2	Instance Variables . . . . .	8
<b>8</b>	<b>Module gift</b>	<b>9</b>
8.1	Variables . . . . .	9
8.2	Class Gift . . . . .	9
8.2.1	Methods . . . . .	9
<b>9</b>	<b>Module girl</b>	<b>10</b>
9.1	Variables . . . . .	10
9.2	Class Girl . . . . .	10
9.2.1	Methods . . . . .	10
9.2.2	Instance Variables . . . . .	10
<b>10</b>	<b>Module log</b>	<b>11</b>
10.1	Functions . . . . .	11
10.2	Variables . . . . .	11
<b>11</b>	<b>Module luxury</b>	<b>12</b>
11.1	Variables . . . . .	12
11.2	Class Luxury . . . . .	12
11.2.1	Methods . . . . .	12
<b>12</b>	<b>Module main</b>	<b>13</b>
12.1	Functions . . . . .	13
12.2	Variables . . . . .	13
<b>13</b>	<b>Module miser</b>	<b>14</b>
13.1	Variables . . . . .	14
13.2	Class Miser . . . . .	14
13.2.1	Methods . . . . .	14
13.2.2	Instance Variables . . . . .	14
<b>14</b>	<b>Module normal</b>	<b>15</b>
14.1	Variables . . . . .	15
14.2	Class normal . . . . .	15
14.2.1	Methods . . . . .	15
14.2.2	Instance Variables . . . . .	15
<b>15</b>	<b>Module q5_utility</b>	<b>16</b>
15.1	Functions . . . . .	16
15.2	Variables . . . . .	16
<b>16</b>	<b>Script script-boys_csv</b>	<b>17</b>
<b>17</b>	<b>Script script-classes_No_Name_dot</b>	<b>18</b>
<b>18</b>	<b>Script script-gift_csv</b>	<b>19</b>
<b>19</b>	<b>Script script-girls_csv</b>	<b>20</b>
<b>20</b>	<b>Script script-packages_No_Name_dot</b>	<b>21</b>

---

<b>21 Module utility</b>	<b>22</b>
21.1 Variables . . . . .	22
21.2 Class Utility . . . . .	22
21.2.1 Methods . . . . .	22

# 1 Module boy

## 1.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

## 1.2 Class Boy

**Known Subclasses:** `geek.Geek`, `generous.Generous`, `miser.Miser`

constructor for class Boy

### 1.2.1 Methods

<code>__init__(self, name_id, attr, bgt, intlg, min_attr, t)</code>
<code>constraint(self, main_bgt, attr)</code>
<code>set_happiness(self, happiness)</code>
<code>set_gf(self, gf)</code>
<code>mod_bgt(self, bgt)</code>

### 1.2.2 Instance Variables

Name	Description
<code>happiness</code>	set girlfriend of boy
<code>gf</code>	modify boy budget if he can't afford gift in gift basket

## 2 Module choosy

### 2.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 2.2 Class choosy



#### 2.2.1 Methods

<code>__init__(self, name_id, attr, main_bgt, intlg, t)</code>
Overrides: girl.Girl. <code>__init__</code>

*Inherited from girl.Girl(Section 9.2)*

`constraint()`, `set_bf()`, `set_happiness()`

#### 2.2.2 Instance Variables

Name	Description
<i>Inherited from girl.Girl (Section 9.2)</i>	
<code>bf</code>	

### 3 Module couple

#### 3.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

#### 3.2 Class Couple

constructor for class Couple having object of boy and girl as attribute to make couple

##### 3.2.1 Methods

<code>__init__(self, b, g)</code>
-----------------------------------

<code>set_happiness(self)</code>
----------------------------------

<code>set_compatibility(self)</code>
--------------------------------------

##### 3.2.2 Instance Variables

Name	Description
<code>g</code>	define and set level of happiness among couples
<code>c_hap</code>	define and set compatibility among couples

## 4 Module desperate

### 4.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 4.2 Class desperate

girl.Girl —  
desperate.desperate

#### 4.2.1 Methods

<code>__init__(self, name_id, attr, main_bgt, intlg, t)</code>
Overrides: girl.Girl.__init__

*Inherited from girl.Girl(Section 9.2)*

`constraint()`, `set_bf()`, `set_happiness()`

#### 4.2.2 Instance Variables

Name	Description
<i>Inherited from girl.Girl (Section 9.2)</i>	
<code>bf</code>	

## 5 Module essential

### 5.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 5.2 Class Essential

gift.Gift —  
essential.Essential

#### 5.2.1 Methods

<code>__init__(self, name_id, price, val, type_)</code>
constructor
Overrides: gift.Gift.__init__



## 6 Module geek

### 6.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 6.2 Class Geek

boy.Boy —  
           geek.Geek  
 constructor

#### 6.2.1 Methods

<code>__init__(self, name_id, atr, bgt, intlg, min_attr, t)</code> Overrides: boy.Boy.__init__
---

*Inherited from boy.Boy(Section 1.2)*

`constraint()`, `mod_bgt()`, `set_gf()`, `set_happiness()`

#### 6.2.2 Instance Variables

Name	Description
<i>Inherited from boy.Boy (Section 1.2)</i> <code>gf</code> , <code>happiness</code>	

## 7 Module generous

### 7.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 7.2 Class Generous

boy.Boy —  
           **generous.Generous**  
 constructor

#### 7.2.1 Methods

<code>__init__(self, name_id, atr, bgt, intlg, min_attr, t)</code> Overrides: boy.Boy.__init__
---

*Inherited from boy.Boy (Section 1.2)*

`constraint()`, `mod_bgt()`, `set_gf()`, `set_happiness()`

#### 7.2.2 Instance Variables

Name	Description
<i>Inherited from boy.Boy (Section 1.2)</i> <code>gf</code> , <code>happiness</code>	

## 8 Module gift

### 8.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 8.2 Class Gift

**Known Subclasses:** `essential.Essential`, `luxury.Luxury`, `utility.Utility`

Constructor for class gift

#### 8.2.1 Methods

<code>__init__(self, name_id, price, val, type_)</code>
---

## 9 Module girl

### 9.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 9.2 Class Girl

**Known Subclasses:** choosy.choosy, desperate.desperate, normal.normal

constructor for class girl

#### 9.2.1 Methods

<code>__init__(self, name_id, attr, main_bgt, intlg, t)</code>
--

<code>constraint(self, bgt)</code>
------------------------------------

<code>set_bf(self, bf)</code>
-------------------------------

<code>set_happiness(self, happiness)</code>
---

#### 9.2.2 Instance Variables

Name	Description
bf	set happiness

## 10 Module log

### 10.1 Functions

<code>log_maker(<i>write</i>)</code>
--------------------------------------

### 10.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

## 11 Module *luxury*

### 11.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> <code>None</code>

### 11.2 Class *Luxury*

`gift.Gift` └─  
           `luxury.Luxury`

#### 11.2.1 Methods

<code>__init__(self, name_id, price, val, type_, rating, eco)</code>
constructor
Overrides: <code>gift.Gift.__init__</code>

## 12 Module main

### 12.1 Functions

**hap\_cpl**( $L$ )

```
A = sorted(L, key=lambda x: x.compat_status, reverse=True)
    print ('
Compatible couples:')
    for i in range(k):
        print (A[i].b.name_id + ' <-> ' + A[i].g.name_id)
        sort list H in descending order according to compatibility t
```

**calc\_hap**( $L$ )

**generous**( $GT, i$ )

**miser**( $GT, i$ )

**geek**( $GT, i$ )

**details**( $L$ )

**display**( $G$ )

**make\_cpl**( $C$ )

**mk\_cpl**()

### 12.2 Variables

Name	Description
__package__	<b>Value:</b> None

## 13 Module miser

### 13.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 13.2 Class Miser

```
boy.Boy └─
          miser.Miser
```

#### 13.2.1 Methods

<code>__init__(self, name_id, atr, bgt, intlg, min_attr, t)</code> Overrides: boy.Boy. <code>__init__</code>
---

*Inherited from boy.Boy(Section 1.2)*

`constraint()`, `mod_bgt()`, `set_gf()`, `set_happiness()`

#### 13.2.2 Instance Variables

Name	Description
<i>Inherited from boy.Boy (Section 1.2)</i> <code>gf</code> , <code>happiness</code>	



## 14 Module normal

### 14.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 14.2 Class normal

girl.Girl —  
           **normal.normal**

#### 14.2.1 Methods

<code>__init__(self, name_id, attr, main_bgt, intlg, t)</code>
constructor
Overrides: girl.Girl.__init__

*Inherited from girl.Girl(Section 9.2)*

constraint(), set\_bf(), set\_happiness()

#### 14.2.2 Instance Variables

Name	Description
<i>Inherited from girl.Girl (Section 9.2)</i>	
bf	

## 15 Module q5\_utility

### 15.1 Functions

<b>utility()</b>
boy type

<b>make_csv(<i>name</i>, <i>list_</i>)</b>
--

### 15.2 Variables

Name	Description
__package__	<b>Value:</b> None

## 16 Script script-boys\_\_csv

## **17   Script script-classes\_\_No\_\_Name\_\_dot**

## 18 Script script-gift\_\_csv

## 19 Script script-girls\_\_csv

## 20 Script script-packages\_No\_Name\_dot

## 21 Module utility

### 21.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 21.2 Class Utility



#### 21.2.1 Methods

<b><code>__init__</code></b> ( <i>self</i> , <i>name_id</i> , <i>price</i> , <i>val</i> , <i>type__</i> , <i>uval</i> , <i>uclass</i> )
constructor
Overrides: <code>gift.Gift.__init__</code>



## Index

- boy (*module*), 2
  - boy.Boy (*class*), 2
    - boy.Boy.\_\_init\_\_ (*method*), 2
    - boy.Boy.constraint (*method*), 2
    - boy.Boy.mod\_bgt (*method*), 2
    - boy.Boy.set\_gf (*method*), 2
    - boy.Boy.set\_happiness (*method*), 2
- choosy (*module*), 3
  - choosy.choosy (*class*), 3
- couple (*module*), 4
  - couple.Couple (*class*), 4
    - couple.Couple.\_\_init\_\_ (*method*), 4
    - couple.Couple.set\_compatibility (*method*), 4
    - couple.Couple.set\_happiness (*method*), 4
- desperate (*module*), 5
  - desperate.desperate (*class*), 5
- essential (*module*), 6
  - essential.Essential (*class*), 6
- geek (*module*), 7
  - geek.Geek (*class*), 7
- generous (*module*), 8
  - generous.Generous (*class*), 8
- gift (*module*), 9
  - gift.Gift (*class*), 9
    - gift.Gift.\_\_init\_\_ (*method*), 9
- girl (*module*), 10
  - girl.Girl (*class*), 10
    - girl.Girl.\_\_init\_\_ (*method*), 10
    - girl.Girl.constraint (*method*), 10
    - girl.Girl.set\_bf (*method*), 10
    - girl.Girl.set\_happiness (*method*), 10
- log (*module*), 11
  - log.log\_maker (*function*), 11
- luxury (*module*), 12
  - luxury.Luxury (*class*), 12
- main (*module*), 13
  - main.calc\_hap (*function*), 13
  - main.details (*function*), 13
  - main.display (*function*), 13
  - main.geek (*function*), 13
  - main.generous (*function*), 13
  - main.hap\_cpl (*function*), 13
  - main.make\_cpl (*function*), 13
  - main.miser (*function*), 13
  - main.mk\_cpl (*function*), 13
- miser (*module*), 14
  - miser.Miser (*class*), 14
- normal (*module*), 15
  - normal.normal (*class*), 15
- q5\_utility (*module*), 16
  - q5\_utility.make\_csv (*function*), 16
  - q5\_utility.utility (*function*), 16
- script-boys\_csv (*script*), 17
- script-classes\_No\_Name\_dot (*script*), 18
- script-gift\_csv (*script*), 19
- script-girls\_csv (*script*), 20
- script-packages\_No\_Name\_dot (*script*), 21
- utility (*module*), 22
  - utility.Utility (*class*), 22