

# Documentation

## API Documentation

April 9, 2017

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Module boy_geek</b>	<b>2</b>
1.1 Variables . . . . .	2
1.2 Class GeekBoy . . . . .	2
1.2.1 Methods . . . . .	2
<b>2 Module boy_generous</b>	<b>3</b>
2.1 Variables . . . . .	3
2.2 Class GenerousBoy . . . . .	3
2.2.1 Methods . . . . .	3
<b>3 Module boy_miser</b>	<b>4</b>
3.1 Variables . . . . .	4
3.2 Class MiserBoy . . . . .	4
3.2.1 Methods . . . . .	4
<b>4 Module boys</b>	<b>5</b>
4.1 Variables . . . . .	5
4.2 Class Boy . . . . .	5
4.2.1 Methods . . . . .	5
<b>5 Module couple</b>	<b>6</b>
5.1 Variables . . . . .	6
5.2 Class Couple . . . . .	6
5.2.1 Methods . . . . .	6
<b>6 Module gift_essential</b>	<b>7</b>
6.1 Variables . . . . .	7
6.2 Class EssentialGift . . . . .	7
6.2.1 Methods . . . . .	7
<b>7 Module gift_luxury</b>	<b>8</b>
7.1 Variables . . . . .	8
7.2 Class LuxuryGift . . . . .	8
7.2.1 Methods . . . . .	8
<b>8 Module gift_utility</b>	<b>9</b>

8.1	Variables . . . . .	9
8.2	Class UtilityGift . . . . .	9
8.2.1	Methods . . . . .	9
<b>9</b>	<b>Module gifts</b>	<b>10</b>
9.1	Variables . . . . .	10
9.2	Class Gift . . . . .	10
9.2.1	Methods . . . . .	10
<b>10</b>	<b>Module girl_choosy</b>	<b>11</b>
10.1	Variables . . . . .	11
10.2	Class ChoosyGirl . . . . .	11
10.2.1	Methods . . . . .	11
<b>11</b>	<b>Module girl_desperate</b>	<b>12</b>
11.1	Variables . . . . .	12
11.2	Class DesperateGirl . . . . .	12
11.2.1	Methods . . . . .	12
<b>12</b>	<b>Module girl_normal</b>	<b>13</b>
12.1	Variables . . . . .	13
12.2	Class NormalGirl . . . . .	13
12.2.1	Methods . . . . .	13
<b>13</b>	<b>Module girls</b>	<b>14</b>
13.1	Variables . . . . .	14
13.2	Class Girl . . . . .	14
13.2.1	Methods . . . . .	14
<b>14</b>	<b>Module q5_driver</b>	<b>15</b>
14.1	Functions . . . . .	15
14.2	Variables . . . . .	15
<b>15</b>	<b>Module utility</b>	<b>16</b>
15.1	Functions . . . . .	16
15.2	Variables . . . . .	16

# 1 Module boy\_geek

## 1.1 Variables

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

## 1.2 Class GeekBoy

boys.Boy —  
           boy\_geek.GeekBoy

boy class for geek boys

### 1.2.1 Methods

<code>__init__(self, name, atr, gfbudget, intelli, min_atr_req, type)</code>
constructor
Overrides: boys.Boy.__init__

*Inherited from boys.Boy(Section 4.2)*

is\_elligible()

## 2 Module *boy\_generous*

### 2.1 Variables

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

### 2.2 Class *GenerousBoy*

```

boys.Boy └─
           boy_generous.GenerousBoy

```

boy class for generous boys

#### 2.2.1 Methods

<code>__init__</code> ( <i>self</i> , <i>name</i> , <i>atr</i> , <i>gfbudget</i> , <i>intelli</i> , <i>min_atr_req</i> , <i>type</i> )
constructor
Overrides: <code>boys.Boy.__init__</code>

*Inherited from boys.Boy(Section 4.2)*

`is_elligible()`

### 3 Module boy\_miser

#### 3.1 Variables

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

#### 3.2 Class MiserBoy

```

boys.Boy └─
           boy_miser.MiserBoy

```

boy class for miser boys

##### 3.2.1 Methods

<code>__init__(self, name, atr, gfbudget, intelli, min_atr_req, type)</code>
constructor
Overrides: boys.Boy.__init__

*Inherited from boys.Boy(Section 4.2)*

`is_elligible()`

## 4 Module boys

### 4.1 Variables

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

### 4.2 Class Boy

**Known Subclasses:** `boy_geek.GeekBoy`, `boy_generous.GenerousBoy`, `boy_miser.MiserBoy`  
 boy class for all boys

#### 4.2.1 Methods

<code>__init__(self, name, atr, gfbudget, intelli, min_atr_req, type)</code>
constructor

<code>is_elligible(self, mbudget, atr)</code>
checks the elligibility of a given Girl, for the current instance of Boy class

## 5 Module couple

### 5.1 Variables

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

### 5.2 Class Couple

couple class for all couples

#### 5.2.1 Methods

<code>__init__(self, boy, girl)</code>
constructor

<code>set_happiness(self)</code>
set the happiness of a couple

<code>set_compatibility(self)</code>
set the compatibility of a couple

## 6 Module `gift_essential`

### 6.1 Variables

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

### 6.2 Class `EssentialGift`

`gifts.Gift` └─  
           **`gift_essential.EssentialGift`**

gift class for essential gifts

#### 6.2.1 Methods

<b><code>__init__</code></b> ( <i>self, name, price, value, type</i> )
constructor
Overrides: <code>gifts.Gift.__init__</code>



## 7 Module `gift_luxury`

### 7.1 Variables

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

### 7.2 Class `LuxuryGift`

`gifts.Gift` —  
`gift_luxury.LuxuryGift`

gift class for luxury gifts

#### 7.2.1 Methods

<code>__init__</code> ( <i>self, name, price, value, type, lxry_rtng, difficulty</i> )
constructor
Overrides: <code>gifts.Gift.__init__</code>

## 8 Module `gift_utility`

### 8.1 Variables

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

### 8.2 Class `UtilityGift`

`gifts.Gift` —  
`gift_utility.UtilityGift`

gift class for utility gifts

#### 8.2.1 Methods

<code>__init__</code> ( <i>self</i> , <i>name</i> , <i>price</i> , <i>value</i> , <i>type</i> , <i>utlty_value</i> , <i>utlty_class</i> )
constructor
Overrides: <code>gifts.Gift.__init__</code>

## 9 Module gifts

### 9.1 Variables

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

### 9.2 Class Gift

**Known Subclasses:** `gift_essential.EssentialGift`, `gift_luxury.LuxuryGift`, `gift_utility.UtilityGift`  
 gift class for all gifts

#### 9.2.1 Methods

<code>__init__</code> ( <i>self</i> , <i>name</i> , <i>price</i> , <i>value</i> , <i>type</i> )
constructor

## 10 Module girl\_choosy

### 10.1 Variables

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

### 10.2 Class ChoosyGirl

girls.Girl —  
                     girl\_choosy.ChoosyGirl

girl class for choosy girls

#### 10.2.1 Methods

<code>__init__(self, name, atr, mbudget, intelli, type)</code>
constructor
Overrides: girls.Girl.__init__

*Inherited from girls.Girl(Section 13.2)*

`is_elligible()`

## 11 Module `girl_desperate`

### 11.1 Variables

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

### 11.2 Class `DesperateGirl`

girls.Girl —  
                     girl\_desperate.DesperateGirl

girl class for desperate girls

#### 11.2.1 Methods

<code>__init__(self, name, atr, mbudget, intelli, type)</code>
constructor
Overrides: girls.Girl.__init__

*Inherited from girls.Girl(Section 13.2)*

`is_elligible()`

## 12 Module girl\_normal

### 12.1 Variables

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

### 12.2 Class NormalGirl

girls.Girl —  
                     girl\_normal.NormalGirl

girl class for normal girls

#### 12.2.1 Methods

<code>__init__</code> ( <i>self, name, atr, mbudget, intelli, type</i> )
constructor
Overrides: girls.Girl. <code>__init__</code>

*Inherited from girls.Girl(Section 13.2)*

`is_elligible()`

## 13 Module girls

### 13.1 Variables

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

### 13.2 Class Girl

**Known Subclasses:** `girl_choosy.ChoosyGirl`, `girl_desperate.DesperateGirl`, `girl_normal.NormalGirl`  
 girl class for all girls

#### 13.2.1 Methods

<code>__init__(self, name, atr, mbudget, intelli, type)</code>
constructor

<code>is_elligible(self, gfbudget)</code>
checks the elligibility of a given Boy, for the current instance of Girl class

## 14 Module q5\_driver

### 14.1 Functions

#### **allocate()**

reads and stores the input from the boys.csv and girls.csv files and then makes the valid couples

#### **calculate\_happiness( $C$ )**

reads and stores the inputs from the gifts.csv file and provide gift exchanges between the couples

#### **set\_girl\_happiness( $c, v1, v2$ )**

sets the happiness of a girl according to her type

#### **hp\_miser( $GFT, c$ )**

provides gifting logic for Miser type Boys and sets the Happiness of the committed Boy and the whole couple, also sets the Compatibility of the couple

#### **hp\_generous( $GFT, c$ )**

provides gifting logic for Generous type Boys and sets the Happiness of the committed Boy and the whole couple, also sets the Compatibility of the couple

#### **hp\_geek( $GFT, c$ )**

provides gifting logic for Geek type Boys and sets the Happiness of the committed Boy and the whole couple, also sets the Compatibility of the couple

#### **print\_hc( $C, k$ )**

prints the k most Happy Couples and k most Compatible Couples

### 14.2 Variables

Name	Description
__package__	Value: None



## 15 Module utility

### 15.1 Functions

<b>utility()</b>
creates the input csv files

<b>create</b> ( <i>file</i> , <i>list</i> )
writes to csv files

### 15.2 Variables

Name	Description
__package__	Value: None

## Index

- boy\_geek (*module*), 2
  - boy\_geek.GeekBoy (*class*), 2
- boy\_generous (*module*), 3
  - boy\_generous.GenerousBoy (*class*), 3
- boy\_miser (*module*), 4
  - boy\_miser.MiserBoy (*class*), 4
- boys (*module*), 5
  - boys.Boy (*class*), 5
    - boys.Boy.\_\_init\_\_ (*method*), 5
    - boys.Boy.is\_elligible (*method*), 5
- couple (*module*), 6
  - couple.Couple (*class*), 6
    - couple.Couple.\_\_init\_\_ (*method*), 6
    - couple.Couple.set\_compatibility (*method*), 6
    - couple.Couple.set\_happiness (*method*), 6
- gift\_essential (*module*), 7
  - gift\_essential.EssentialGift (*class*), 7
- gift\_luxury (*module*), 8
  - gift\_luxury.LuxuryGift (*class*), 8
- gift\_utility (*module*), 9
  - gift\_utility.UtilityGift (*class*), 9
- gifts (*module*), 10
  - gifts.Gift (*class*), 10
    - gifts.Gift.\_\_init\_\_ (*method*), 10
- girl\_choosy (*module*), 11
  - girl\_choosy.ChoosyGirl (*class*), 11
- girl\_desperate (*module*), 12
  - girl\_desperate.DesperateGirl (*class*), 12
- girl\_normal (*module*), 13
  - girl\_normal.NormalGirl (*class*), 13
- girls (*module*), 14
  - girls.Girl (*class*), 14
    - girls.Girl.\_\_init\_\_ (*method*), 14
    - girls.Girl.is\_elligible (*method*), 14
- q5\_driver (*module*), 15
  - q5\_driver.allocate (*function*), 15
  - q5\_driver.calculate\_happiness (*function*), 15
  - q5\_driver.hp\_geek (*function*), 15
  - q5\_driver.hp\_generous (*function*), 15
  - q5\_driver.hp\_miser (*function*), 15
  - q5\_driver.print\_hc (*function*), 15
  - q5\_driver.set\_girl\_happiness (*function*), 15
  - utility (*module*), 16
    - utility.create (*function*), 16
    - utility.utility (*function*), 16