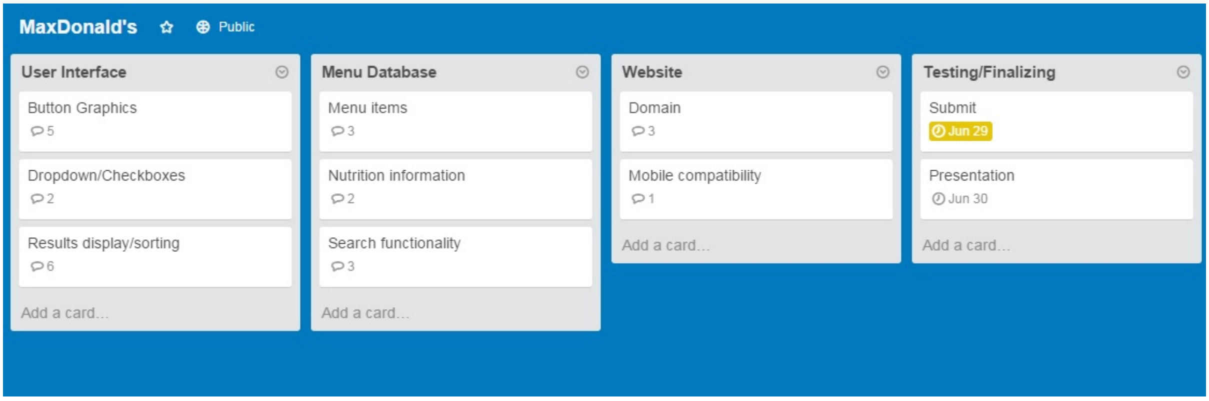
**Title:** MaxDonald’s

**Who:** Ian Moore, Ethan Fellows, Samuel Carnes, Hussain Alqudaihi

**Project Tracker:** https://trello.com/b/x49MSwHo/maxdonald-s



MaxDonald's ☆ Public

User Interface
- Button Graphics 💬 5
- Dropdown/Checkboxes 💬 2
- Results display/sorting 💬 6
- Add a card...

Menu Database
- Menu items 💬 3
- Nutrition information 💬 2
- Search functionality 💬 3
- Add a card...

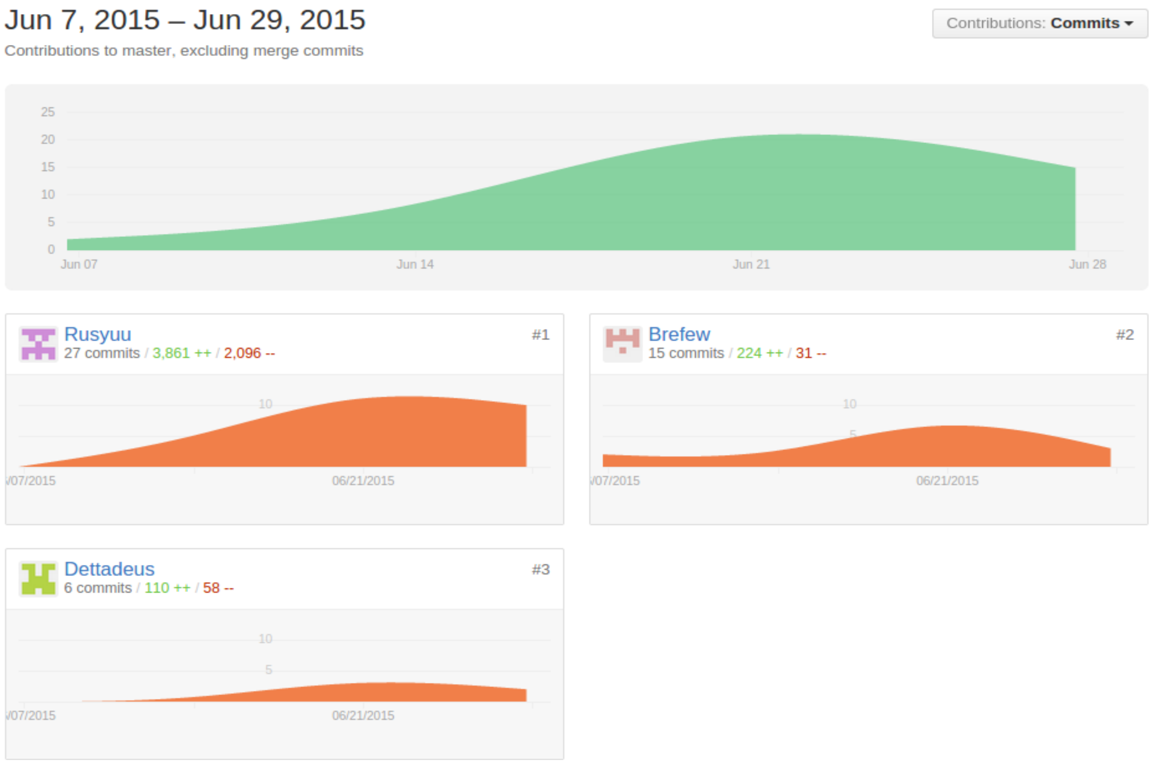
Website
- Domain 💬 3
- Mobile compatibility 💬 1
- Add a card...

Testing/Finalizing
- Submit Jun 29
- Presentation Jun 30
- Add a card...

**VCS:** https://github.com/Brefew/MaxDonalds

**Video:** Video can be found in the MaxDonalds github folder, as MaxDonalds_video.wmv

**Screenshot:**



Jun 7, 2015 – Jun 29, 2015

Contributions to master, excluding merge commits

Contributions: **Commits**

Rusyuu #1
27 commits / 3,861 ++ / 2,096 --

Brefew #2
15 commits / 224 ++ / 31 --

Dettadeus #3
6 commits / 110 ++ / 58 --

***Hussain is not shown in the Github charts, however the commit log shows that he was present and active during the whole project.**

**Deployment:** After cloning the files from github, open a new terminal. Go to the necessary directory, and run the python –m CGIHTTPServer 8000 command. Open a webpage, and go to localhost:8000/ and type the path and name of the file to access the website. (cgi-bin/chickenRun.py, etc.)

# MaxDonald's

Generated by Doxygen 1.8.6

Tue Jun 30 2015 05:36:58

# Contents

# Chapter 1

# Namespace Index

## 1.1 Packages

Here are the packages with brief descriptions (if available):

# Chapter 2

# File Index

## 2.1 File List

Here is a list of all files with brief descriptions:

# Chapter 3

# Namespace Documentation

## 3.1 burgerRun Namespace Reference

Explanation of this particular module.

### Functions

- def byteify

  *Byteify documentation.*

### Variables

- tuple response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple stats = byteify(json.load(response))

  *stats documentation*
- dictionary drink = {}

  *drink dictionary documentation*
- dictionary burger = {}

  *burger dictionary documentation*
- dictionary chicken = {}

  *chicken dictionary documentation*
- dictionary salad = {}

  *salad dictionary documentation*
- dictionary dessert = {}

  *dessert dictionary documentation*
- dictionary fries = {}

  *fry dictionary documentation*
- int drCn = 1

  *drCn Documentation*
- int buCn = 1

  *buCn Documentation*
- int chCn = 1

  *chCn Documentation*
- int saCn = 1

  *saCn Documentation*
- int deCn = 1

*deCn Documentation*

- int frCn = 1

  *frCn Documentation*
- tuple r = len(stats)

  *len documentation*
- int usr_money = 529

  *usr_money documentation*
- budget = usr_money

  *budget documentation*
- minmoney = budget

  *minmoney documentation*
- int price = 0

  *price documentation*
- int calpercent = 0

  *calpercent documentation*
- int run1 = 0

  *run1 documentation*
- string item = ""

  *item documentation*
- list order = [ ]

  *order documentation*
- dictionary work_dict = {}

  *work_dict documentation*

## 3.1.1 Detailed Description

Explanation of this particular module. Runs a test based on the burger category

## 3.1.2 Function Documentation

### 3.1.2.1 def burgerRun.byteify ( *input* )

Byteify documentation.

Referenced from the HW4 function Converts the unicode of JSON keys

Definition at line 73 of file burgerRun.py.

## 3.1.3 Variable Documentation

### 3.1.3.1 int burgerRun.buCn = 1

buCn Documentation

The integer keeping track of the burger count index

Definition at line 124 of file burgerRun.py.

### 3.1.3.2 burgerRun.budget = usr_money

budget documentation

Maintains the usr_money value

Definition at line 557 of file burgerRun.py.

#### 3.1.3.3 dictionary burgerRun.burger = {}

burger dictionary documentation

Slice of the primary dictionary designated for burgers

Definition at line 99 of file burgerRun.py.

#### 3.1.3.4 list burgerRun.calpercent = 0

calpercent documentation

Int for keeping track of the calories of food items per one cent

Definition at line 569 of file burgerRun.py.

#### 3.1.3.5 int burgerRun.chCn = 1

chCn Documentation

The integer keeping track of the chicken count index

Definition at line 128 of file burgerRun.py.

#### 3.1.3.6 dictionary burgerRun.chicken = {}

chicken dictionary documentation

Slice of the primary dictionary designated for chicken

Definition at line 103 of file burgerRun.py.

#### 3.1.3.7 int burgerRun.deCn = 1

deCn Documentation

The integer keeping track of the dessert count index

Definition at line 136 of file burgerRun.py.

#### 3.1.3.8 dictionary burgerRun.dessert = {}

dessert dictionary documentation

Slice of the primary dictionary designated for dessert

Definition at line 111 of file burgerRun.py.

#### 3.1.3.9 int burgerRun.drCn = 1

drCn Documentation

The integer keeping track of the drink count index

Definition at line 120 of file burgerRun.py.

#### 3.1.3.10 dictionary burgerRun.drink = {}

drink dictionary documentation

Slice of the primary dictionary designated for drinks

Definition at line 95 of file burgerRun.py.

#### 3.1.3.11 int burgerRun.frCn = 1

frCn Documentation

The integer keeping track of the fry count index

Definition at line 140 of file burgerRun.py.

#### 3.1.3.12 dictionary burgerRun.fries = {}

fry dictionary documentation

Slice of the primary dictionary designated for fries

Definition at line 115 of file burgerRun.py.

#### 3.1.3.13 burgerRun.item = ""

item documentation

An empty variable that takes the name of an item assigned later

Definition at line 577 of file burgerRun.py.

#### 3.1.3.14 list burgerRun.minmoney = budget

minmoney documentation

Maintains the budget value

Definition at line 561 of file burgerRun.py.

#### 3.1.3.15 list burgerRun.order = [ ]

order documentation

Empty list to be filled with items that pass the tests for the items with the optimal calorie-per-cent rate

Definition at line 581 of file burgerRun.py.

#### 3.1.3.16 list burgerRun.price = 0

price documentation

Standin price variable for the upcoming calculations

Definition at line 565 of file burgerRun.py.

#### 3.1.3.17 tuple burgerRun.r = len(stats)

len documentation

A value to keep track of the stats max length in a for loop that splits it up

Definition at line 147 of file burgerRun.py.

#### 3.1.3.18 tuple burgerRun.response = urllib2.urlopen("file:MaxDonalds.json")

response documentation

Variable for holding the json input file

Definition at line 86 of file burgerRun.py.

#### 3.1.3.19 int burgerRun.run1 = 0

run1 documentation

Acts as a toggle to keep track of the upcoming while loop

Definition at line 573 of file burgerRun.py.

#### 3.1.3.20 int burgerRun.saCn = 1

saCn Documentation

The integer keeping track of the salad count index

Definition at line 132 of file burgerRun.py.

#### 3.1.3.21 dictionary burgerRun.salad = {}

salad dictionary documentation

Slice of the primary dictionary designated for salads

Definition at line 107 of file burgerRun.py.

#### 3.1.3.22 tuple burgerRun.stats = byteify(json.load(response))

stats documentation

A variable that acts as the converted json folder, having passed through byteify

Definition at line 90 of file burgerRun.py.

#### 3.1.3.23 int burgerRun.usr_money = 529

usr_money documentation

A test value to apply to the code (displayed in cents)

Definition at line 553 of file burgerRun.py.

#### 3.1.3.24 dictionary burgerRun.work_dict = {}

work_dict documentation

An empty dictionary that will handle the order[] list to be filled

Definition at line 585 of file burgerRun.py.

## 3.2 chickenRun Namespace Reference

Explanation of this particular module.

## Functions

- def byteify

  *Byteify documentation.*

## Variables

- tuple response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple stats = byteify(json.load(response))

  *stats documentation*
- dictionary drink = {}

  *drink dictionary documentation*
- dictionary burger = {}

  *burger dictionary documentation*
- dictionary chicken = {}

  *chicken dictionary documentation*
- dictionary salad = {}

  *salad dictionary documentation*
- dictionary dessert = {}

  *dessert dictionary documentation*
- dictionary fries = {}

  *fry dictionary documentation*
- int drCn = 1

  *drCn Documentation*
- int buCn = 1

  *buCn Documentation*
- int chCn = 1

  *chCn Documentation*
- int saCn = 1

  *saCn Documentation*
- int deCn = 1

  *deCn Documentation*
- int frCn = 1

  *frCn Documentation*
- tuple r = len(stats)

  *len documentation*
- int usr_money = 529

  *usr_money documentation*
- budget = usr_money

  *budget documentation*
- minmoney = budget

  *minmoney documentation*
- int price = 0

  *price documentation*
- int calpercent = 0

  *calpercent documentation*
- int run1 = 0

  *run1 documentation*
- string item = ""

  *item documentation*

- list order = [ ]

    *order documentation*
- dictionary work_dict = {}

    *work_dict documentation*

## 3.2.1 Detailed Description

Explanation of this particular module. Runs a test based on the chicken category

## 3.2.2 Function Documentation

### 3.2.2.1 def chickenRun.byteify ( *input* )

Byteify documentation.

Referenced from the HW4 function Converts the unicode of JSON keys

Definition at line 73 of file chickenRun.py.

## 3.2.3 Variable Documentation

### 3.2.3.1 int chickenRun.buCn = 1

buCn Documentation

The integer keeping track of the burger count index

Definition at line 124 of file chickenRun.py.

### 3.2.3.2 chickenRun.budget = usr_money

budget documentation

Maintains the usr_money value

Definition at line 557 of file chickenRun.py.

### 3.2.3.3 dictionary chickenRun.burger = {}

burger dictionary documentation

Slice of the primary dictionary designated for burgers

Definition at line 99 of file chickenRun.py.

### 3.2.3.4 list chickenRun.calpercent = 0

calpercent documentation

Int for keeping track of the calories of food items per one cent

Definition at line 569 of file chickenRun.py.

### 3.2.3.5 int chickenRun.chCn = 1

chCn Documentation

The integer keeping track of the chicken count index

Definition at line 128 of file chickenRun.py.

#### 3.2.3.6 dictionary chickenRun.chicken = {}

chicken dictionary documentation

Slice of the primary dictionary designated for chicken

Definition at line 103 of file chickenRun.py.

#### 3.2.3.7 int chickenRun.deCn = 1

deCn Documentation

The integer keeping track of the dessert count index

Definition at line 136 of file chickenRun.py.

#### 3.2.3.8 dictionary chickenRun.dessert = {}

dessert dictionary documentation

Slice of the primary dictionary designated for dessert

Definition at line 111 of file chickenRun.py.

#### 3.2.3.9 int chickenRun.drCn = 1

drCn Documentation

The integer keeping track of the drink count index

Definition at line 120 of file chickenRun.py.

#### 3.2.3.10 dictionary chickenRun.drink = {}

drink dictionary documentation

Slice of the primary dictionary designated for drinks

Definition at line 95 of file chickenRun.py.

#### 3.2.3.11 int chickenRun.frCn = 1

frCn Documentation

The integer keeping track of the fry count index

Definition at line 140 of file chickenRun.py.

#### 3.2.3.12 dictionary chickenRun.fries = {}

fry dictionary documentation

Slice of the primary dictionary designated for fries

Definition at line 115 of file chickenRun.py.

#### 3.2.3.13 chickenRun.item = ""

item documentation

An empty variable that takes the name of an item assigned later

Definition at line 577 of file chickenRun.py.

#### 3.2.3.14 list chickenRun.minmoney = budget

minmoney documentation

Maintains the budget value

Definition at line 561 of file chickenRun.py.

#### 3.2.3.15 list chickenRun.order = [ ]

order documentation

Empty list to be filled with items that pass the tests for the items with the optimal calorie-per-cent rate

Definition at line 581 of file chickenRun.py.

#### 3.2.3.16 list chickenRun.price = 0

price documentation

Standin price variable for the upcoming calculations

Definition at line 565 of file chickenRun.py.

#### 3.2.3.17 tuple chickenRun.r = len(stats)

len documentation

A value to keep track of the stats max length in a for loop that splits it up

Definition at line 147 of file chickenRun.py.

#### 3.2.3.18 tuple chickenRun.response = urllib2.urlopen("file:MaxDonalds.json")

response documentation

Variable for holding the json input file

Definition at line 86 of file chickenRun.py.

#### 3.2.3.19 int chickenRun.run1 = 0

run1 documentation

Acts as a toggle to keep track of the upcoming while loop

Definition at line 573 of file chickenRun.py.

#### 3.2.3.20 int chickenRun.saCn = 1

saCn Documentation

The integer keeping track of the salad count index

Definition at line 132 of file chickenRun.py.

#### 3.2.3.21 dictionary chickenRun.salad = {}

salad dictionary documentation

Slice of the primary dictionary designated for salads

Definition at line 107 of file chickenRun.py.

#### 3.2.3.22 tuple chickenRun.stats = byteify(json.load(response))

stats documentation

A variable that acts as the converted json folder, having passed through byteify

Definition at line 90 of file chickenRun.py.

#### 3.2.3.23 int chickenRun.usr_money = 529

usr_money documentation

A test value to apply to the code (displayed in cents)

Definition at line 553 of file chickenRun.py.

#### 3.2.3.24 dictionary chickenRun.work_dict = {}

work_dict documentation

An empty dictionary that will handle the order[] list to be filled

Definition at line 585 of file chickenRun.py.

## 3.3 dessertRun Namespace Reference

Explanation of this particular module.

### Functions

- def byteify

  *Byteify documentation.*

### Variables

- tuple response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple stats = byteify(json.load(response))

  *stats documentation*
- dictionary drink = {}

  *drink dictionary documentation*
- dictionary burger = {}

  *burger dictionary documentation*
- dictionary chicken = {}

*chicken dictionary documentation*

- dictionary salad = {}

  *salad dictionary documentation*
- dictionary dessert = {}

  *dessert dictionary documentation*
- dictionary fries = {}

  *fry dictionary documentation*
- int drCn = 1

  *drCn Documentation*
- int buCn = 1

  *buCn Documentation*
- int chCn = 1

  *chCn Documentation*
- int saCn = 1

  *saCn Documentation*
- int deCn = 1

  *deCn Documentation*
- int frCn = 1

  *frCn Documentation*
- tuple r = len(stats)

  *len documentation*
- int usr_money = 529

  *usr_money documentation*
- budget = usr_money

  *budget documentation*
- minmoney = budget

  *minmoney documentation*
- int price = 0

  *price documentation*
- int calpercent = 0

  *calpercent documentation*
- int run1 = 0

  *run1 documentation*
- string item = ""

  *item documentation*
- list order = [ ]

  *order documentation*
- dictionary work_dict = {}

  *work_dict documentation*

## 3.3.1 Detailed Description

Explanation of this particular module. Runs a test based on the dessert category

## 3.3.2 Function Documentation

### 3.3.2.1 def dessertRun.byteify ( *input* )

Byteify documentation.

Referenced from the HW4 function Converts the unicode of JSON keys

Definition at line 73 of file dessertRun.py.

### 3.3.3 Variable Documentation

#### 3.3.3.1 int dessertRun.buCn = 1

buCn Documentation

The integer keeping track of the burger count index

Definition at line 124 of file dessertRun.py.

#### 3.3.3.2 dessertRun.budget = usr_money

budget documentation

Maintains the usr_money value

Definition at line 557 of file dessertRun.py.

#### 3.3.3.3 dictionary dessertRun.burger = {}

burger dictionary documentation

Slice of the primary dictionary designated for burgers

Definition at line 99 of file dessertRun.py.

#### 3.3.3.4 list dessertRun.calpercent = 0

calpercent documentation

Int for keeping track of the calories of food items per one cent

Definition at line 569 of file dessertRun.py.

#### 3.3.3.5 int dessertRun.chCn = 1

chCn Documentation

The integer keeping track of the chicken count index

Definition at line 128 of file dessertRun.py.

#### 3.3.3.6 dictionary dessertRun.chicken = {}

chicken dictionary documentation

Slice of the primary dictionary designated for chicken

Definition at line 103 of file dessertRun.py.

#### 3.3.3.7 int dessertRun.deCn = 1

deCn Documentation

The integer keeping track of the dessert count index

Definition at line 136 of file dessertRun.py.

#### 3.3.3.8 dictionary dessertRun.dessert = {}

dessert dictionary documentation

Slice of the primary dictionary designated for dessert

Definition at line 111 of file dessertRun.py.

#### 3.3.3.9 int dessertRun.drCn = 1

drCn Documentation

The integer keeping track of the drink count index

Definition at line 120 of file dessertRun.py.

#### 3.3.3.10 dictionary dessertRun.drink = {}

drink dictionary documentation

Slice of the primary dictionary designated for drinks

Definition at line 95 of file dessertRun.py.

#### 3.3.3.11 int dessertRun.frCn = 1

frCn Documentation

The integer keeping track of the fry count index

Definition at line 140 of file dessertRun.py.

#### 3.3.3.12 dictionary dessertRun.fries = {}

fry dictionary documentation

Slice of the primary dictionary designated for fries

Definition at line 115 of file dessertRun.py.

#### 3.3.3.13 dessertRun.item = ""

item documentation

An empty variable that takes the name of an item assigned later

Definition at line 577 of file dessertRun.py.

#### 3.3.3.14 list dessertRun.minmoney = budget

minmoney documentation

Maintains the budget value

Definition at line 561 of file dessertRun.py.

#### 3.3.3.15 list dessertRun.order = [ ]

order documentation

Empty list to be filled with items that pass the tests for the items with the optimal calorie-per-cent rate

Definition at line 581 of file dessertRun.py.

#### 3.3.3.16 list dessertRun.price = 0

price documentation

Standin price variable for the upcoming calculations

Definition at line 565 of file dessertRun.py.

#### 3.3.3.17 tuple dessertRun.r = len(stats)

len documentation

A value to keep track of the stats max length in a for loop that splits it up

Definition at line 147 of file dessertRun.py.

#### 3.3.3.18 tuple dessertRun.response = urllib2.urlopen("file:MaxDonalds.json")

response documentation

Variable for holding the json input file

Definition at line 86 of file dessertRun.py.

#### 3.3.3.19 int dessertRun.run1 = 0

run1 documentation

Acts as a toggle to keep track of the upcoming while loop

Definition at line 573 of file dessertRun.py.

#### 3.3.3.20 int dessertRun.saCn = 1

saCn Documentation

The integer keeping track of the salad count index

Definition at line 132 of file dessertRun.py.

#### 3.3.3.21 dictionary dessertRun.salad = {}

salad dictionary documentation

Slice of the primary dictionary designated for salads

Definition at line 107 of file dessertRun.py.

#### 3.3.3.22 tuple dessertRun.stats = byteify(json.load(response))

stats documentation

A variable that acts as the converted json folder, having passed through byteify

Definition at line 90 of file dessertRun.py.

#### 3.3.3.23 int dessertRun.usr_money = 529

usr_money documentation

A test value to apply to the code (displayed in cents)

Definition at line 553 of file dessertRun.py.

#### 3.3.3.24 dictionary dessertRun.work_dict = {}

work_dict documentation

An empty dictionary that will handle the order[] list to be filled

Definition at line 585 of file dessertRun.py.

## 3.4 drinksRun Namespace Reference

Explanation of this particular module.

### Functions

- def byteify

  *Byteify documentation.*

### Variables

- tuple response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple stats = byteify(json.load(response))

  *stats documentation*
- dictionary drink = {}

  *drink dictionary documentation*
- dictionary burger = {}

  *burger dictionary documentation*
- dictionary chicken = {}

  *chicken dictionary documentation*
- dictionary salad = {}

  *salad dictionary documentation*
- dictionary dessert = {}

  *dessert dictionary documentation*
- dictionary fries = {}

  *fry dictionary documentation*
- int drCn = 1

  *drCn Documentation*
- int buCn = 1

  *buCn Documentation*
- int chCn = 1

  *chCn Documentation*
- int saCn = 1

  *saCn Documentation*
- int deCn = 1

*deCn Documentation*

- int frCn = 1

  *frCn Documentation*
- tuple r = len(stats)

  *len documentation*
- int usr_money = 529

  *usr_money documentation*
- budget = usr_money

  *budget documentation*
- minmoney = budget

  *minmoney documentation*
- int price = 0

  *price documentation*
- int calpercent = 0

  *calpercent documentation*
- int run1 = 0

  *run1 documentation*
- string item = ""

  *item documentation*
- list order = [ ]

  *order documentation*
- dictionary work_dict = {}

  *work_dict documentation*

## 3.4.1 Detailed Description

Explanation of this particular module. Runs a test based on the drink category

## 3.4.2 Function Documentation

### 3.4.2.1 def drinksRun.byteify ( *input* )

Byteify documentation.

Referenced from the HW4 function Converts the unicode of JSON keys

Definition at line 73 of file drinksRun.py.

## 3.4.3 Variable Documentation

### 3.4.3.1 int drinksRun.buCn = 1

buCn Documentation

The integer keeping track of the burger count index

Definition at line 124 of file drinksRun.py.

### 3.4.3.2 drinksRun.budget = usr_money

budget documentation

Maintains the usr_money value

Definition at line 557 of file drinksRun.py.

#### 3.4.3.3 dictionary drinksRun.burger = {}

burger dictionary documentation

Slice of the primary dictionary designated for burgers

Definition at line 99 of file drinksRun.py.

#### 3.4.3.4 list drinksRun.calpercent = 0

calpercent documentation

Int for keeping track of the calories of food items per one cent

Definition at line 569 of file drinksRun.py.

#### 3.4.3.5 int drinksRun.chCn = 1

chCn Documentation

The integer keeping track of the chicken count index

Definition at line 128 of file drinksRun.py.

#### 3.4.3.6 dictionary drinksRun.chicken = {}

chicken dictionary documentation

Slice of the primary dictionary designated for chicken

Definition at line 103 of file drinksRun.py.

#### 3.4.3.7 int drinksRun.deCn = 1

deCn Documentation

The integer keeping track of the dessert count index

Definition at line 136 of file drinksRun.py.

#### 3.4.3.8 dictionary drinksRun.dessert = {}

dessert dictionary documentation

Slice of the primary dictionary designated for dessert

Definition at line 111 of file drinksRun.py.

#### 3.4.3.9 int drinksRun.drCn = 1

drCn Documentation

The integer keeping track of the drink count index

Definition at line 120 of file drinksRun.py.

#### 3.4.3.10 dictionary drinksRun.drink = {}

drink dictionary documentation

Slice of the primary dictionary designated for drinks

Definition at line 95 of file drinksRun.py.

#### 3.4.3.11 int drinksRun.frCn = 1

frCn Documentation

The integer keeping track of the fry count index

Definition at line 140 of file drinksRun.py.

#### 3.4.3.12 dictionary drinksRun.fries = {}

fry dictionary documentation

Slice of the primary dictionary designated for fries

Definition at line 115 of file drinksRun.py.

#### 3.4.3.13 drinksRun.item = ""

item documentation

An empty variable that takes the name of an item assigned later

Definition at line 577 of file drinksRun.py.

#### 3.4.3.14 list drinksRun.minmoney = budget

minmoney documentation

Maintains the budget value

Definition at line 561 of file drinksRun.py.

#### 3.4.3.15 list drinksRun.order = [ ]

order documentation

Empty list to be filled with items that pass the tests for the items with the optimal calorie-per-cent rate

Definition at line 581 of file drinksRun.py.

#### 3.4.3.16 list drinksRun.price = 0

price documentation

Standin price variable for the upcoming calculations

Definition at line 565 of file drinksRun.py.

#### 3.4.3.17 tuple drinksRun.r = len(stats)

len documentation

A value to keep track of the stats max length in a for loop that splits it up

Definition at line 147 of file drinksRun.py.

#### 3.4.3.18 tuple drinksRun.response = urllib2.urlopen("file:MaxDonalds.json")

response documentation

Variable for holding the json input file

Definition at line 86 of file drinksRun.py.

#### 3.4.3.19 int drinksRun.run1 = 0

run1 documentation

Acts as a toggle to keep track of the upcoming while loop

Definition at line 573 of file drinksRun.py.

#### 3.4.3.20 int drinksRun.saCn = 1

saCn Documentation

The integer keeping track of the salad count index

Definition at line 132 of file drinksRun.py.

#### 3.4.3.21 dictionary drinksRun.salad = {}

salad dictionary documentation

Slice of the primary dictionary designated for salads

Definition at line 107 of file drinksRun.py.

#### 3.4.3.22 tuple drinksRun.stats = byteify(json.load(response))

stats documentation

A variable that acts as the converted json folder, having passed through byteify

Definition at line 90 of file drinksRun.py.

#### 3.4.3.23 int drinksRun.usr_money = 529

usr_money documentation

A test value to apply to the code (displayed in cents)

Definition at line 553 of file drinksRun.py.

#### 3.4.3.24 dictionary drinksRun.work_dict = {}

work_dict documentation

An empty dictionary that will handle the order[] list to be filled

Definition at line 585 of file drinksRun.py.

## 3.5 friesRun Namespace Reference

Explanation of this particular module.

## Functions

- def byteify

  *Byteify documentation.*

## Variables

- tuple response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple stats = byteify(json.load(response))

  *stats documentation*
- dictionary drink = {}

  *drink dictionary documentation*
- dictionary burger = {}

  *burger dictionary documentation*
- dictionary chicken = {}

  *chicken dictionary documentation*
- dictionary salad = {}

  *salad dictionary documentation*
- dictionary dessert = {}

  *dessert dictionary documentation*
- dictionary fries = {}

  *fry dictionary documentation*
- int drCn = 1

  *drCn Documentation*
- int buCn = 1

  *buCn Documentation*
- int chCn = 1

  *chCn Documentation*
- int saCn = 1

  *saCn Documentation*
- int deCn = 1

  *deCn Documentation*
- int frCn = 1

  *frCn Documentation*
- tuple r = len(stats)

  *len documentation*
- int usr_money = 529

  *usr_money documentation*
- budget = usr_money

  *budget documentation*
- minmoney = budget

  *minmoney documentation*
- int price = 0

  *price documentation*
- int calpercent = 0

  *calpercent documentation*
- int run1 = 0

  *run1 documentation*
- string item = ""

  *item documentation*

- list order = [ ]

  *order documentation*
- dictionary work_dict = {}

  *work_dict documentation*

## 3.5.1 Detailed Description

Explanation of this particular module. Runs a test based on the fry category

## 3.5.2 Function Documentation

### 3.5.2.1 def friesRun.byteify ( *input* )

Byteify documentation.

Referenced from the HW4 function Converts the unicode of JSON keys

Definition at line 73 of file friesRun.py.

## 3.5.3 Variable Documentation

### 3.5.3.1 int friesRun.buCn = 1

buCn Documentation

The integer keeping track of the burger count index

Definition at line 124 of file friesRun.py.

### 3.5.3.2 friesRun.budget = usr_money

budget documentation

Maintains the usr_money value

Definition at line 557 of file friesRun.py.

### 3.5.3.3 dictionary friesRun.burger = {}

burger dictionary documentation

Slice of the primary dictionary designated for burgers

Definition at line 99 of file friesRun.py.

### 3.5.3.4 list friesRun.calpercent = 0

calpercent documentation

Int for keeping track of the calories of food items per one cent

Definition at line 569 of file friesRun.py.

### 3.5.3.5 int friesRun.chCn = 1

chCn Documentation

The integer keeping track of the chicken count index

Definition at line 128 of file friesRun.py.

#### 3.5.3.6 dictionary friesRun.chicken = {}

chicken dictionary documentation

Slice of the primary dictionary designated for chicken

Definition at line 103 of file friesRun.py.

#### 3.5.3.7 int friesRun.deCn = 1

deCn Documentation

The integer keeping track of the dessert count index

Definition at line 136 of file friesRun.py.

#### 3.5.3.8 dictionary friesRun.dessert = {}

dessert dictionary documentation

Slice of the primary dictionary designated for dessert

Definition at line 111 of file friesRun.py.

#### 3.5.3.9 int friesRun.drCn = 1

drCn Documentation

The integer keeping track of the drink count index

Definition at line 120 of file friesRun.py.

#### 3.5.3.10 dictionary friesRun.drink = {}

drink dictionary documentation

Slice of the primary dictionary designated for drinks

Definition at line 95 of file friesRun.py.

#### 3.5.3.11 int friesRun.frCn = 1

frCn Documentation

The integer keeping track of the fry count index

Definition at line 140 of file friesRun.py.

#### 3.5.3.12 dictionary friesRun.fries = {}

fry dictionary documentation

Slice of the primary dictionary designated for fries

Definition at line 115 of file friesRun.py.

#### 3.5.3.13 friesRun.item = ""

item documentation

An empty variable that takes the name of an item assigned later

Definition at line 577 of file friesRun.py.

#### 3.5.3.14 list friesRun.minmoney = budget

minmoney documentation

Maintains the budget value

Definition at line 561 of file friesRun.py.

#### 3.5.3.15 list friesRun.order = [ ]

order documentation

Empty list to be filled with items that pass the tests for the items with the optimal calorie-per-cent rate

Definition at line 581 of file friesRun.py.

#### 3.5.3.16 list friesRun.price = 0

price documentation

Standin price variable for the upcoming calculations

Definition at line 565 of file friesRun.py.

#### 3.5.3.17 tuple friesRun.r = len(stats)

len documentation

A value to keep track of the stats max length in a for loop that splits it up

Definition at line 147 of file friesRun.py.

#### 3.5.3.18 tuple friesRun.response = urllib2.urlopen("file:MaxDonalds.json")

response documentation

Variable for holding the json input file

Definition at line 86 of file friesRun.py.

#### 3.5.3.19 int friesRun.run1 = 0

run1 documentation

Acts as a toggle to keep track of the upcoming while loop

Definition at line 573 of file friesRun.py.

#### 3.5.3.20 int friesRun.saCn = 1

saCn Documentation

The integer keeping track of the salad count index

Definition at line 132 of file friesRun.py.

#### 3.5.3.21 dictionary friesRun.salad = {}

salad dictionary documentation

Slice of the primary dictionary designated for salads

Definition at line 107 of file friesRun.py.

#### 3.5.3.22 tuple friesRun.stats = **byteify**(json.load(**response**))

stats documentation

A variable that acts as the converted json folder, having passed through byteify

Definition at line 90 of file friesRun.py.

#### 3.5.3.23 int friesRun.usr_money = 529

usr_money documentation

A test value to apply to the code (displayed in cents)

Definition at line 553 of file friesRun.py.

#### 3.5.3.24 dictionary friesRun.work_dict = {}

work_dict documentation

An empty dictionary that will handle the order[] list to be filled

Definition at line 585 of file friesRun.py.

## 3.6 saladRun Namespace Reference

Explanation of this particular module.

### Functions

- def byteify

  *Byteify documentation.*

### Variables

- tuple response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple stats = byteify(json.load(response))

  *stats documentation*
- dictionary drink = {}

  *drink dictionary documentation*
- dictionary burger = {}

  *burger dictionary documentation*
- dictionary chicken = {}

*chicken dictionary documentation*

- dictionary salad = {}

  *salad dictionary documentation*

- dictionary dessert = {}

  *dessert dictionary documentation*

- dictionary fries = {}

  *fry dictionary documentation*

- int drCn = 1

  *drCn Documentation*

- int buCn = 1

  *buCn Documentation*

- int chCn = 1

  *chCn Documentation*

- int saCn = 1

  *saCn Documentation*

- int deCn = 1

  *deCn Documentation*

- int frCn = 1

  *frCn Documentation*

- tuple r = len(stats)

  *len documentation*

- int usr_money = 529

  *usr_money documentation*

- budget = usr_money

  *budget documentation*

- minmoney = budget

  *minmoney documentation*

- int price = 0

  *price documentation*

- int calpercent = 0

  *calpercent documentation*

- int run1 = 0

  *run1 documentation*

- string item = ""

  *item documentation*

- list order = [ ]

  *order documentation*

- dictionary work_dict = {}

  *work_dict documentation*

## 3.6.1 Detailed Description

Explanation of this particular module. Runs a test based on the salad category

## 3.6.2 Function Documentation

### 3.6.2.1 def saladRun.byteify ( *input* )

Byteify documentation.

Referenced from the HW4 function Converts the unicode of JSON keys

Definition at line 73 of file saladRun.py.

### 3.6.3 Variable Documentation

#### 3.6.3.1 int saladRun.buCn = 1

buCn Documentation

The integer keeping track of the burger count index

Definition at line 124 of file saladRun.py.

#### 3.6.3.2 saladRun.budget = usr_money

budget documentation

Maintains the usr_money value

Definition at line 557 of file saladRun.py.

#### 3.6.3.3 dictionary saladRun.burger = {}

burger dictionary documentation

Slice of the primary dictionary designated for burgers

Definition at line 99 of file saladRun.py.

#### 3.6.3.4 list saladRun.calpercent = 0

calpercent documentation

Int for keeping track of the calories of food items per one cent

Definition at line 569 of file saladRun.py.

#### 3.6.3.5 int saladRun.chCn = 1

chCn Documentation

The integer keeping track of the chicken count index

Definition at line 128 of file saladRun.py.

#### 3.6.3.6 dictionary saladRun.chicken = {}

chicken dictionary documentation

Slice of the primary dictionary designated for chicken

Definition at line 103 of file saladRun.py.

#### 3.6.3.7 int saladRun.deCn = 1

deCn Documentation

The integer keeping track of the dessert count index

Definition at line 136 of file saladRun.py.

#### 3.6.3.8 dictionary saladRun.dessert = {}

dessert dictionary documentation

Slice of the primary dictionary designated for dessert

Definition at line 111 of file saladRun.py.

#### 3.6.3.9 int saladRun.drCn = 1

drCn Documentation

The integer keeping track of the drink count index

Definition at line 120 of file saladRun.py.

#### 3.6.3.10 dictionary saladRun.drink = {}

drink dictionary documentation

Slice of the primary dictionary designated for drinks

Definition at line 95 of file saladRun.py.

#### 3.6.3.11 int saladRun.frCn = 1

frCn Documentation

The integer keeping track of the fry count index

Definition at line 140 of file saladRun.py.

#### 3.6.3.12 dictionary saladRun.fries = {}

fry dictionary documentation

Slice of the primary dictionary designated for fries

Definition at line 115 of file saladRun.py.

#### 3.6.3.13 saladRun.item = ""

item documentation

An empty variable that takes the name of an item assigned later

Definition at line 577 of file saladRun.py.

#### 3.6.3.14 list saladRun.minmoney = budget

minmoney documentation

Maintains the budget value

Definition at line 561 of file saladRun.py.

#### 3.6.3.15 list saladRun.order = [ ]

order documentation

Empty list to be filled with items that pass the tests for the items with the optimal calorie-per-cent rate

Definition at line 581 of file saladRun.py.

#### 3.6.3.16 list saladRun.price = 0

price documentation

Standin price variable for the upcoming calculations

Definition at line 565 of file saladRun.py.

#### 3.6.3.17 tuple saladRun.r = len(stats)

len documentation

A value to keep track of the stats max length in a for loop that splits it up

Definition at line 147 of file saladRun.py.

#### 3.6.3.18 tuple saladRun.response = urllib2.urlopen("file:MaxDonalds.json")

response documentation

Variable for holding the json input file

Definition at line 86 of file saladRun.py.

#### 3.6.3.19 int saladRun.run1 = 0

run1 documentation

Acts as a toggle to keep track of the upcoming while loop

Definition at line 573 of file saladRun.py.

#### 3.6.3.20 int saladRun.saCn = 1

saCn Documentation

The integer keeping track of the salad count index

Definition at line 132 of file saladRun.py.

#### 3.6.3.21 dictionary saladRun.salad = {}

salad dictionary documentation

Slice of the primary dictionary designated for salads

Definition at line 107 of file saladRun.py.

#### 3.6.3.22 tuple saladRun.stats = byteify(json.load(response))

stats documentation

A variable that acts as the converted json folder, having passed through byteify

Definition at line 90 of file saladRun.py.

#### 3.6.3.23 int saladRun.usr_money = 529

usr_money documentation

A test value to apply to the code (displayed in cents)

Definition at line 553 of file saladRun.py.

#### 3.6.3.24 dictionary saladRun.work_dict = {}

work_dict documentation

An empty dictionary that will handle the order[] list to be filled

Definition at line 585 of file saladRun.py.

# Chapter 4

# File Documentation

## 4.1 burgerRun.py File Reference

### Namespaces

- burgerRun

  *Explanation of this particular module.*

### Functions

- def burgerRun.byteify

  *Byteify documentation.*

### Variables

- tuple burgerRun.response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple burgerRun.stats = byteify(json.load(response))

  *stats documentation*
- dictionary burgerRun.drink = {}

  *drink dictionary documentation*
- dictionary burgerRun.burger = {}

  *burger dictionary documentation*
- dictionary burgerRun.chicken = {}

  *chicken dictionary documentation*
- dictionary burgerRun.salad = {}

  *salad dictionary documentation*
- dictionary burgerRun.dessert = {}

  *dessert dictionary documentation*
- dictionary burgerRun.fries = {}

  *fry dictionary documentation*
- int burgerRun.drCn = 1

  *drCn Documentation*
- int burgerRun.buCn = 1

  *buCn Documentation*
- int burgerRun.chCn = 1

  *chCn Documentation*

- int burgerRun.saCn = 1

  *saCn Documentation*
- int burgerRun.deCn = 1

  *deCn Documentation*
- int burgerRun.frCn = 1

  *frCn Documentation*
- tuple burgerRun.r = len(stats)

  *len documentation*
- int burgerRun.usr_money = 529

  *usr_money documentation*
- burgerRun.budget = usr_money

  *budget documentation*
- burgerRun.minmoney = budget

  *minmoney documentation*
- int burgerRun.price = 0

  *price documentation*
- int burgerRun.calpercent = 0

  *calpercent documentation*
- int burgerRun.run1 = 0

  *run1 documentation*
- string burgerRun.item = ""

  *item documentation*
- list burgerRun.order = [ ]

  *order documentation*
- dictionary burgerRun.work_dict = {}

  *work_dict documentation*

## 4.2 chickenRun.py File Reference

### Namespaces

- chickenRun

  *Explanation of this particular module.*

### Functions

- def chickenRun.byteify

  *Byteify documentation.*

### Variables

- tuple chickenRun.response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple chickenRun.stats = byteify(json.load(response))

  *stats documentation*
- dictionary chickenRun.drink = {}

  *drink dictionary documentation*
- dictionary chickenRun.burger = {}

  *burger dictionary documentation*

- dictionary chickenRun.chicken = {}

  *chicken dictionary documentation*
- dictionary chickenRun.salad = {}

  *salad dictionary documentation*
- dictionary chickenRun.dessert = {}

  *dessert dictionary documentation*
- dictionary chickenRun.fries = {}

  *fry dictionary documentation*
- int chickenRun.drCn = 1

  *drCn Documentation*
- int chickenRun.buCn = 1

  *buCn Documentation*
- int chickenRun.chCn = 1

  *chCn Documentation*
- int chickenRun.saCn = 1

  *saCn Documentation*
- int chickenRun.deCn = 1

  *deCn Documentation*
- int chickenRun.frCn = 1

  *frCn Documentation*
- tuple chickenRun.r = len(stats)

  *len documentation*
- int chickenRun.usr_money = 529

  *usr_money documentation*
- chickenRun.budget = usr_money

  *budget documentation*
- chickenRun.minmoney = budget

  *minmoney documentation*
- int chickenRun.price = 0

  *price documentation*
- int chickenRun.calpercent = 0

  *calpercent documentation*
- int chickenRun.run1 = 0

  *run1 documentation*
- string chickenRun.item = ""

  *item documentation*
- list chickenRun.order = [ ]

  *order documentation*
- dictionary chickenRun.work_dict = {}

  *work_dict documentation*

## 4.3 dessertRun.py File Reference

### Namespaces

- dessertRun

  *Explanation of this particular module.*

## Functions

- def dessertRun.byteify

    *Byteify documentation.*

## Variables

- tuple dessertRun.response = urllib2.urlopen("file:MaxDonalds.json")

    *response documentation*
- tuple dessertRun.stats = byteify(json.load(response))

    *stats documentation*
- dictionary dessertRun.drink = {}

    *drink dictionary documentation*
- dictionary dessertRun.burger = {}

    *burger dictionary documentation*
- dictionary dessertRun.chicken = {}

    *chicken dictionary documentation*
- dictionary dessertRun.salad = {}

    *salad dictionary documentation*
- dictionary dessertRun.dessert = {}

    *dessert dictionary documentation*
- dictionary dessertRun.fries = {}

    *fry dictionary documentation*
- int dessertRun.drCn = 1

    *drCn Documentation*
- int dessertRun.buCn = 1

    *buCn Documentation*
- int dessertRun.chCn = 1

    *chCn Documentation*
- int dessertRun.saCn = 1

    *saCn Documentation*
- int dessertRun.deCn = 1

    *deCn Documentation*
- int dessertRun.frCn = 1

    *frCn Documentation*
- tuple dessertRun.r = len(stats)

    *len documentation*
- int dessertRun.usr_money = 529

    *usr_money documentation*
- dessertRun.budget = usr_money

    *budget documentation*
- dessertRun.minmoney = budget

    *minmoney documentation*
- int dessertRun.price = 0

    *price documentation*
- int dessertRun.calpercent = 0

    *calpercent documentation*
- int dessertRun.run1 = 0

    *run1 documentation*
- string dessertRun.item = ""

    *item documentation*

- list dessertRun.order = [ ]

  *order documentation*
- dictionary dessertRun.work_dict = {}

  *work_dict documentation*

## 4.4 drinksRun.py File Reference

### Namespaces

- drinksRun

  *Explanation of this particular module.*

### Functions

- def drinksRun.byteify

  *Byteify documentation.*

### Variables

- tuple drinksRun.response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple drinksRun.stats = byteify(json.load(response))

  *stats documentation*
- dictionary drinksRun.drink = {}

  *drink dictionary documentation*
- dictionary drinksRun.burger = {}

  *burger dictionary documentation*
- dictionary drinksRun.chicken = {}

  *chicken dictionary documentation*
- dictionary drinksRun.salad = {}

  *salad dictionary documentation*
- dictionary drinksRun.dessert = {}

  *dessert dictionary documentation*
- dictionary drinksRun.fries = {}

  *fry dictionary documentation*
- int drinksRun.drCn = 1

  *drCn Documentation*
- int drinksRun.buCn = 1

  *buCn Documentation*
- int drinksRun.chCn = 1

  *chCn Documentation*
- int drinksRun.saCn = 1

  *saCn Documentation*
- int drinksRun.deCn = 1

  *deCn Documentation*
- int drinksRun.frCn = 1

  *frCn Documentation*
- tuple drinksRun.r = len(stats)

  *len documentation*

- int drinksRun.usr_money = 529

  *usr_money documentation*
- drinksRun.budget = usr_money

  *budget documentation*
- drinksRun.minmoney = budget

  *minmoney documentation*
- int drinksRun.price = 0

  *price documentation*
- int drinksRun.calpercent = 0

  *calpercent documentation*
- int drinksRun.run1 = 0

  *run1 documentation*
- string drinksRun.item = ""

  *item documentation*
- list drinksRun.order = [ ]

  *order documentation*
- dictionary drinksRun.work_dict = {}

  *work_dict documentation*

## 4.5 friesRun.py File Reference

### Namespaces

- friesRun

  *Explanation of this particular module.*

### Functions

- def friesRun.byteify

  *Byteify documentation.*

### Variables

- tuple friesRun.response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple friesRun.stats = byteify(json.load(response))

  *stats documentation*
- dictionary friesRun.drink = {}

  *drink dictionary documentation*
- dictionary friesRun.burger = {}

  *burger dictionary documentation*
- dictionary friesRun.chicken = {}

  *chicken dictionary documentation*
- dictionary friesRun.salad = {}

  *salad dictionary documentation*
- dictionary friesRun.dessert = {}

  *dessert dictionary documentation*
- dictionary friesRun.fries = {}

  *fry dictionary documentation*

- int friesRun.drCn = 1

  *drCn Documentation*
- int friesRun.buCn = 1

  *buCn Documentation*
- int friesRun.chCn = 1

  *chCn Documentation*
- int friesRun.saCn = 1

  *saCn Documentation*
- int friesRun.deCn = 1

  *deCn Documentation*
- int friesRun.frCn = 1

  *frCn Documentation*
- tuple friesRun.r = len(stats)

  *len documentation*
- int friesRun.usr_money = 529

  *usr_money documentation*
- friesRun.budget = usr_money

  *budget documentation*
- friesRun.minmoney = budget

  *minmoney documentation*
- int friesRun.price = 0

  *price documentation*
- int friesRun.calpercent = 0

  *calpercent documentation*
- int friesRun.run1 = 0

  *run1 documentation*
- string friesRun.item = ""

  *item documentation*
- list friesRun.order = [ ]

  *order documentation*
- dictionary friesRun.work_dict = {}

  *work_dict documentation*

## 4.6 saladRun.py File Reference

### Namespaces

- saladRun

  *Explanation of this particular module.*

### Functions

- def saladRun.byteify

  *Byteify documentation.*

## Variables

- tuple saladRun.response = urllib2.urlopen("file:MaxDonalds.json")

  *response documentation*
- tuple saladRun.stats = byteify(json.load(response))

  *stats documentation*
- dictionary saladRun.drink = {}

  *drink dictionary documentation*
- dictionary saladRun.burger = {}

  *burger dictionary documentation*
- dictionary saladRun.chicken = {}

  *chicken dictionary documentation*
- dictionary saladRun.salad = {}

  *salad dictionary documentation*
- dictionary saladRun.dessert = {}

  *dessert dictionary documentation*
- dictionary saladRun.fries = {}

  *fry dictionary documentation*
- int saladRun.drCn = 1

  *drCn Documentation*
- int saladRun.buCn = 1

  *buCn Documentation*
- int saladRun.chCn = 1

  *chCn Documentation*
- int saladRun.saCn = 1

  *saCn Documentation*
- int saladRun.deCn = 1

  *deCn Documentation*
- int saladRun.frCn = 1

  *frCn Documentation*
- tuple saladRun.r = len(stats)

  *len documentation*
- int saladRun.usr_money = 529

  *usr_money documentation*
- saladRun.budget = usr_money

  *budget documentation*
- saladRun.minmoney = budget

  *minmoney documentation*
- int saladRun.price = 0

  *price documentation*
- int saladRun.calpercent = 0

  *calpercent documentation*
- int saladRun.run1 = 0

  *run1 documentation*
- string saladRun.item = ""

  *item documentation*
- list saladRun.order = [ ]

  *order documentation*
- dictionary saladRun.work_dict = {}

  *work_dict documentation*

# Index