

GIT TASKS

Obtendo ajuda

- Linha de comando (exemplo: `git help`, `git/gl cmd -h`, `man git-cmd`);
- Ajuda online (exemplo: Google, StackOverflow...);
- Por favor, não use seu celular ou outras fontes (estamos tentando trackear o que você precisa).

Sobre as tasks

- Cada sessão tem um total de 7 tasks com uma série de pequenos passos;
- Para cada passo, vamos indicar como o repositório resultante deve ser;
- Se você cometer um erro, não se preocupe! Tente consertar o problema e lembre que você pode pesquisar na web;
- Não volte para uma task já feita: quando você passar para próxima task, assumimos que a task anterior foi completada (se você finalizou todos os passos) ou abandonada;
- Antes de você executar uma task, você deve executar o comando `ut-run` com o número da próxima task a ser executada. Esse script vai limpar o que você fez e resetar o repositório para o estado que ele deveria estar antes da próxima task ser feita.

Intervalos

- Você poderá fazer intervalos entre as tasks. Você saberá quando isso pode acontecer quando ver o símbolo:



Informações

- Qual o seu GitHub?
- Senha sudo: ?

Fit-cli

- Em cada sessão você irá trabalhar como parte do time da Fit Industries Inc., desenvolvedores do fit-cli. Fit-cli é uma ferramenta de linha de comando que, dado seu gênero, idade, peso e altura, computa uma estimativa do número de calorias que você queima por dia apenas por estar vivo (também conhecido como taxa de metabolismo basal).
- Sempre quis ter calorias queimadas sem deixar a linha de comando? Este app é para você! Junto com Alice (a CEO) e seu time você vai mudar o mundo.

Git

- O time do fit-cli são grandes fãs do Git e fazem uso dele no dia-a-dia. Nesta sessão, você vai utilizar o Git para localizar mudanças no código fonte. Fit-cli é escrito em Python, mas você não precisa saber como programar em Python. Vamos dizer exatamente o que você deve mudar e como fazer isso. Tudo o que você precisa descobrir é como usar o Git para completar as tasks.



Você pode fazer uma pausa agora, se você quiser.

A seguir: task 0

TASK 0

Clonando o repositório do fit-cli

Vamos preparar a sua máquina para você estar pronto para contribuir com o fit-cli. Para fazer isso, vamos clonar o repositório do fit-cli.

1. Abra o terminal, navegue até ~/Desktop e clone o repositório <https://github.com/helenaandrade/fit-cli> - [check](#)



Você pode fazer uma pausa agora, se você quiser.

Deixe seu repositório pronto para próxima task executando o comando `sudo run.sh git 1`

A seguir: task 1/6

TASK 1

Adicionado o README

Talvez você não tenha percebido, mas nosso app não tem um arquivo readme. Vamos consertar isso.

1. Crie o README.md com [esse conteúdo](#);
2. Rastreie o README.md - [check](#);
3. Vamos adicionar instruções de uso ao readme. Abra o README.md e faça [essas modificações](#);
4. Crie um commit com todas as modificações em README.md e a mensagem “added readme” - [check](#).



Você pode fazer uma pausa agora, se você quiser.

Deixe seu repositório pronto para próxima task executando o comando `sudo run.sh git 2`

A seguir: task 2/6

TASK 2

Adicionando o peso em quilos

O fato de que os usuários apenas podem adicionar seu peso em libras e sua altura em polegadas se tornou um grande problema para usuários do fit-cli que não são americanos. Alice instruiu você a implementar duas novas features: a permissão para que os usuários insiram o seu peso em quilos e a sua altura em metros.

Primeiro, vamos implementar a feature relacionada ao peso.

1. Vamos ser organizados e trabalhar em uma branch separada. Crie uma nova branch `feat/kilos` e mude para ela - [check](#);
2. Abra o arquivo `fit.py` e faça [essas mudanças](#) - [check](#);
3. Crie um commit com as mudanças em `fit.py` e a mensagem “kilos feature” - [check](#);
4. Bob (o CTO) vai revisar suas mudanças. Execute o comando `pr_kilos_send.sh` para enviar uma pull request;
5. Bob enviou alguns comentários sobre o seu código. Primeiro, ele sugeriu que nós adicionássemos uma mensagem de ajuda a flag quilos. Isso é fácil de consertar! Abra o `fit.py` e faça [essas mudanças](#) - [check](#);

Bob também sugeriu que ao invés de usar `w_const_female` e `w_const_male`, nós deveríamos mudar para um dicionário. Isso vai levar mais tempo para consertar, então vamos deixar de lado e trabalhar na feature da altura.



Você pode fazer uma pausa agora, se você quiser.

Deixe seu repositório pronto para próxima task executando o comando `sudo run.sh git 3`

A seguir: task 3/6

TASK 3

Adicionando altura em metros (3.1)

3.1 Vamos começar com a feature para adicionar a altura em metros.

1. Crie uma nova branch `feat/meters` que diverge da `master` e mude para `feat/meters` - [check](#)
2. Abra o arquivo `fit.py` e faça [esta mudança](#) [-check](#)

TASK 3

Adicionando altura em metros (3.2)

3.2 Alice acabou de entrar no seu escritório! Um dos principais usuários do fit-cli reportou um bug: nenhum erro é reportado ao usuário ele se ele inserir um valor diferente de “female” ou “male” para o gênero. Como ela quer que você conserte isso imediatamente, nós vamos consertar este bug e depois voltaremos a trabalhar nas nossas features.

1. Volte para `master` - [check](#)
2. Abra o arquivo `fit.py` e faça [esta mudança](#) - [check](#)
3. Crie um commit com as mudanças em `fit.py` e a mensagem “fix error checking bug” [-check](#)
4. Publique as mudanças

Ufa! O bug foi consertado. Vamos voltar a trabalhar nas nossas features.



Você pode fazer uma pausa agora, se você quiser.

Deixe seu repositório pronto para próxima task executando o comando `sudo run.sh git 4`

A seguir: task 4/6

TASK 4

Juntando as features (4.1)

4.1 Bob quer juntar as features para que estejam prontas para sair na próxima release. Então, ao invés de voltar a trabalhar na feature da altura em metros, vamos deixar isso de lado e terminar a feature do peso em quilos.

1. Volte para `feat/kilos` - [check](#)
2. Vamos adicionar o dicionário como Bob queria. Abra o arquivo `fit.py` e faça [esta mudança](#) -[check](#)
3. Crie um commit com as mudanças em `fit.py` e a mensagem “kilos improvements” -[check](#)
4. Atualize a pull request: execute o comando `pr_kilos_update.sh`

TASK 4

Juntando as features (4.2)

4.2 Enquanto nós esperamos o feedback de Bob sobre a feature do peso, vamos voltar e terminar a função da altura.

1. Volte para `feat/meters` - [check](#)
2. Agora, vamos modificar a computação do RMB para usar metros se o usuário especificar assim. Abra o arquivo `fit.py` e faça [esta mudança](#) -[check](#)
3. Crie um commit com as mudanças em `fit.py` e a mensagem “meters feature” -[check](#)
4. Bob (o CTO) vai revisar suas mudanças: execute o comando `pr_meters_send.sh` para enviar uma pull request



Você pode fazer uma pausa agora, se você quiser.

Deixe seu repositório pronto para próxima task executando o comando `sudo run.sh git 5`

A seguir: task 5/6

TASK 5

Resolvendo conflitos (5.1)

5.1 Bob aceitou seu pedido de merge com a feature do peso, mas agora ele está dizendo que a pull request do peso não foi aplicada de forma limpa e está nos pedindo que resolvamos os conflitos com a master e atualizemos a pull request.

1. Atualize sua branch `master` local com as novas mudanças na master remota - [check](#)
2. Volte para `feat/meters` e faça a junção das mudanças da master (que vai gerar conflitos) - [check](#)
3. Vamos começar a resolver os conflitos. Abra o `fit.py` e faça [esta mudança](#) - [check](#)

TASK 5

Resolvendo conflitos (5.2)

5.2 Bob apareceu no seu escritório! Aparentemente nós usamos o shebang errado todo esse tempo. Ele quer que conserte isso agora. Nós vamos consertar esse bug e depois vamos voltar a resolver os conflitos.

1. Mude para `master` - [check](#)
2. Abra o `fit.py` e faça [esta mudança](#) - [check](#)
3. Crie um commit com as mudanças em `fit.py` e a mensagem “fix shebang” - [check](#)
4. Publique as mudanças

TASK 5

Resolvendo conflitos (5.3)

5.3 Vamos terminar de resolver os conflitos.

1. Mude para `feat/meters` e volte para o estado que estávamos antes de corrigir o bug - [check](#)
2. Abra o `fit.py` e faça [esta mudança](#) - [check](#)
3. Finalize com a operação de fusão - [check](#)
4. Atualize a pull request da feature da altura: execute o comando `pr-meters_update.sh`



Você pode fazer uma pausa agora, se você quiser.

Deixe seu repositório pronto para próxima task executando o comando `sudo run.sh git 6`

A seguir: task 6/6

TASK 6

Organizando o código

6.1 É hora de organizar o código. Esta deve ser uma task bastante pequena, então vamos torná-la simples e trabalhar na master.

1. Mude para `master` e publique as mudanças - [check](#)
2. Abra o `fit.py` e faça [esta mudança](#) - [check](#)
3. Crie um commit com as mudanças em `fit.py` e a mensagem “join const dicts” - [check](#)

TASK 6

Organizando o código

6.2 Depois de uma volta no parque, você percebeu que o último commit pode estar errado. Então vamos tentar algo diferente.

1. Volte para o mesmo estado antes do commit - [check](#)
2. Abra o `fit.py` e faça [esta mudança](#) - [check](#)
3. Crie um commit com a mensagem “switch to using nt for consts” - [check](#)
4. Usar tuplas nomeadas parece melhor do que mesclar os dicionários, então vamos em frente e publique as mudanças

Fim!

Por favor, complete o [formulário do final da sessão do git](#).

Se essa é a sua segunda sessão, por favor complete também o [formulário do final do estudo](#).

Material para Task 0

Voltar para task 0

Command	Expected output
Assuming you are at ~/Documents	
ls	fit-cli
Assuming you are at ~/Documents/fit-cli	
ls	fit.py
git branch	* master
git log --oneline	0e032fd first cut at fit-cli app f821dce Initialize repository

Material para Task 1

Voltar para task 1

fit-cli

=====

The ultimate fitness app. Give your gender, age, weight and height to fit-cli and get an estimate of the number of calories you burn per day just for being alive; all of this without leaving the command line!

Voltar para task 1

Command	Expected output
<code>git status</code>	<pre>On branch master Your branch is up-to-date with 'origin/master'. Changes to be committed: (use "git reset HEAD <file>..." to unstage) new file: README.md</pre>

Voltar para task 1

1) Adicione ao final do arquivo:

To run fit-cli execute the fit.py script which accepts the following optional flags: -g/--gender, -a/--age, -w/--weight, -h/--height.

Voltar para task 1

Command	Expected output
<code>git log -p -1 --oneline</code>	<pre>00f747d added readme diff --git a/README.md b/README.md new file mode 100644 index 0000000..d787c5d --- /dev/null +++ b/README.md @@ -0,0 +1,5 @@ +fit-cli +===== +The ultimate fitness app. Give your gender, age, weight and height to +fit-cli and get an estimate of the number of calories you burn per day +just for being alive; all of this without leaving the command line! + +To run fit-cli execute the fit.py script which accepts the following +optional flags:-g/--gender, -a/--age, -w/--weight, -h/--height.</pre>
<code>git log --oneline</code>	<pre>00f747d added readme f294049 first cut at fit-cli app ef5940e Initialize repository</pre>

Material para Task 2

Voltar para task 2

Command	Expected output
<code>git branch</code>	<pre>* feat/kilos master</pre>

Voltar para task 2

- 1) Depois da linha 13 (depois da height flag), adicione o seguinte texto em uma nova linha:

```
parser.add_argument('-k', '--kilos', action='store_true')
```

- 2) Atualize a chamada da rmb para corresponder:

```
result = rmb(  
args.gender, args.age, args.weight, args.height, use_kg=args.kilos)
```

- 3) Atualize a função rmb para corresponder:

```
def rmb(gender, age, weight, height, use_kg=False):  
    w_const_female = 9.56 if use_kg else 4.35  
    w_const_male = 13.75 if use_kg else 6.25  
    if gender == 'female':  
        return (655 + w_const_female * weight + 4.7 * height - 4.7 * age) *  
1.1  
    else:  
        return (66 + w_const_male * weight + 12.7 * height - 6.76 * age) * 1.1
```

Voltar para task 2

Command	Expected output
git diff	<pre>diff --git a/fit.py b/fit.py index c28ba26..a7e4384 100755 --- a/fit.py +++ b/fit.py @@ -11,18 +11,22 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') + parser.add_argument('-k', '--kilos', action='store_true') args = parser.parse_args() - result = rmb(args.gender, args.age, args.weight, args.height) + result = rmb(+ args.gender, args.age, args.weight, args.height, use_kg=args.kilos) print('Your resting metabolic rate is {}'.format(result)) -def rmb(gender, age, weight, height): +def rmb(gender, age, weight, height, use_kg=False): + w_const_female = 9.56 if use_kg else 4.35 + w_const_male = 13.75 if use_kg else 6.25 if gender == 'female': - return (655 + 4.35 * weight + 4.7 * height - 4.7 * age) * 1.1 + return (655 + w_const_female * weight + 4.7 * height - 4.7 * age) * 1.1 else: - return (66 + 6.25 * weight + 12.7 * height - 6.76 * age) * 1.1 + return (66 + w_const_male * weight + 12.7 * height - 6.76 * age) * 1.1 if __name__ == '__main__':</pre>

Voltar para task 2

Command	Expected output
<code>git log --oneline -p -1</code>	<pre>a7c51f5 kilos feature diff --git a/fit.py b/fit.py index c28ba26..a7e4384 100755 --- a/fit.py +++ b/fit.py @@ -11,18 +11,22 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') + parser.add_argument('-k', '--kilos', action='store_true') args = parser.parse_args() - result = rmb(args.gender, args.age, args.weight, args.height) + result = rmb(+ args.gender, args.age, args.weight, args.height, use_kg=args.kilos) print('Your resting metabolic rate is {}'.format(result)) -def rmb(gender, age, weight, height): +def rmb(gender, age, weight, height, use_kg=False): + w_const_female = 9.56 if use_kg else 4.35 + w_const_male = 13.75 if use_kg else 6.25 if gender == 'female': - return (655 + 4.35 * weight + 4.7 * height - 4.7 * age) * 1.1 + return (655 + w_const_female * weight + 4.7 * height - 4.7 * age) * 1.1 else: - return (66 + 6.25 * weight + 12.7 * height - 6.76 * age) * 1.1 + return (66 + w_const_male * weight + 12.7 * height - 6.76 * age) * 1.1 if __name__ == '__main__':</pre>
<code>git log --oneline</code>	<pre>a7c51f5 kilos feature 00f747d added readme f294049 first cut at fit-cli app ef5940e Initialize repository</pre>

Voltar para task 2

5) Mude o argumento do quilo para:

```
parser.add_argument(  
    '-k', '--kilos', action='store_true',  
    help='if true, then the weight given as input is interpreted as kg')
```


Voltar para task 2

Command	Expected output
<code>git diff</code>	<pre>diff --git a/fit.py b/fit.py index a7e4384..04d6006 100755 --- a/fit.py +++ b/fit.py @@ -11,7 +11,9 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') - parser.add_argument('-k', '--kilos', action='store_true') + parser.add_argument(+ '-k', '--kilos', action='store_true', + help='if true, then the weight given as input is interpreted as kg') args = parser.parse_args() result = rmb(</pre>

Material para Task 3

Voltar para task 3 (3.1)

Command	Expected output
<code>git branch -v</code>	<pre>feat/kilos c34c636 kilos feature * feat/meters c23a513 added readme master c23a513 [ahead 1] added readme</pre>

Voltar para task 3 (3.1)

- 1) Depois da linha 13 (depois da height flag), adicione o seguinte texto em uma nova linha:

```
parser.add_argument('-k', '--kilos', action='store_true',  
help='if true, the height input will be interpreted in meters')
```

- 2) Atualize a chamada da rmb para corresponder:

```
result = rmb(  
    args.gender, args.age, args.weight, args.height, use_m=args.meters)
```

- 3) Atualize a função rmb para corresponder:

```
def rmb(gender, age, weight, height, use_m=False):  
    h_const= {  
        'female': 1.8 if use_m else 4.7,  
        'male': 5 if use_m else 12.7  
    }  
    if gender == 'female':  
        return (655 + 4.35 * weight + 4.7 * height - 4.7 * age) * 1.1  
    else:  
        return (66 + 6.25 * weight + 12.7 * height - 6.76 * age) * 1.1
```

Voltar para task 3 (3.1)

Command	Expected output
git diff	<pre>diff --git a/fit.py b/fit.py index c28ba26..6af4580 100755 --- a/fit.py +++ b/fit.py @@ -11,14 +11,22 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') + parser.add_argument(+ '-m', '--meters', action='store_true', + help='if true, the height input will be interpreted in meters') args = parser.parse_args() - result = rmb(args.gender, args.age, args.weight, args.height) + result = rmb(+ args.gender, args.age, args.weight, args.height, use_m=args.meters) print('Your resting metabolic rate is {}'.format(result)) -def rmb(gender, age, weight, height): +def rmb(gender, age, weight, height, use_m=False): + h_const= { + 'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7 + } + if gender == 'female': + return (655 + 4.35 * weight + 4.7 * height - 4.7 * age) * 1.1 + else:</pre>

Voltar para task 3 (3.2)

Command	Expected output
<code>git branch</code>	<code>feat/kilos</code> <code>feat/meters</code> <code>* master</code>
<code>git status</code>	On branch master Your branch is ahead of 'origin/master' by 1 commit. (use "git push" to publish your local commits) nothing to commit, working directory clean

Voltar para task 3 (3.2)

- 1) Substitua a linha que declara o argumento de gênero (linha 10) por:

```
parser.add_argument('gender', choices=['female', 'male'])
```

Voltar para task 3 (3.2)

Command	Expected output
git diff	<pre>diff --git a/fit.py b/fit.py index c28ba26..29c708c 100755 --- a/fit.py +++ b/fit.py @@ -7,7 +7,7 @@ def main(): parser = argparse.ArgumentParser(description=('fit-cli: compute your resting metabolic rate from the command line')) - parser.add_argument('gender') + parser.add_argument('gender', choices=['female', 'male']) parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches')</pre>

Voltar para task 3 (3.2)

Command	Expected output
<code>git log --oneline -p -1</code>	<pre>9a33c59 fix error checking bug diff --git a/fit.py b/fit.py index c28ba26..29c708c 100755 --- a/fit.py +++ b/fit.py @@ -7,7 +7,7 @@ def main(): parser = argparse.ArgumentParser(description=('fit-cli: compute your resting metabolic rate from the command line')) - parser.add_argument('gender') + parser.add_argument('gender', choices=['female', 'male']) parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches')</pre>
<code>git log --oneline</code>	<pre>9a33c59 fix error checking bug c23a513 added readme 333ed80 first cut at fit-cli app 85e982f Initialize repository</pre>

Material para Task 4

Voltar para task 4 (4.1)

Command	Expected output
git status	<pre>On branch feat/kilos Changes not staged for commit: (use "git add <file>..." to update what will be committed) (use "git checkout -- <file>..." to discard changes in working directory) modified: fit.py no changes added to commit (use "git add" and/or "git commit -a")</pre>
git diff	<pre>diff --git a/fit.py b/fit.py index a7e4384..04d6006 100755 --- a/fit.py +++ b/fit.py @@ -11,7 +11,9 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') - parser.add_argument('-k', '--kilos', action='store_true') + parser.add_argument(+ '-k', '--kilos', action='store_true', + help='if true, then the weight given as input is interpreted as kg') args = parser.parse_args() result = rmb(</pre>

Voltar para task 4 (4.1)

1) Atualize a função rmb para:

```
def rmb(gender, age, weight, height, use_kg=False):  
    w_const = {  
        'female': 9.56 if use_kg else 4.35,  
        'male': 13.75 if use_kg else 6.25  
    }  
    if gender == 'female':  
        return (655 + w_const[gender] * weight + 4.7 * height - 4.7 * age) * 1.1  
    else:  
        return (66 + w_const[gender] * weight + 12.7 * height - 6.76 * age) * 1.1
```

Voltar para task 4 (4.1)

Command	Expected output
<code>git diff</code>	<pre>diff --git a/fit.py b/fit.py index a7e4384..33865f7 100755 --- a/fit.py +++ b/fit.py @@ -11,7 +11,9 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') - parser.add_argument('-k', '--kilos', action='store_true') + parser.add_argument(+ '-k', '--kilos', action='store_true', + help='if true, then the weight given as input is interpreted as kg') args = parser.parse_args() result = rmb(@@ -21,12 +23,14 @@ def main(): def rmb(gender, age, weight, height, use_kg=False): - w_const_female = 9.56 if use_kg else 4.35 - w_const_male = 13.75 if use_kg else 6.25 + w_const = { + 'female': 9.56 if use_kg else 4.35, + 'male': 13.75 if use_kg else 6.25 + } if gender == 'female': - return (655 + w_const_female * weight + 4.7 * height - 4.7 * age) * 1.1 + return (655 + w_const[gender] * weight + 4.7 * height - 4.7 * age) * 1.1 else: - return (66 + w_const_male * weight + 12.7 * height - 6.76 * age) * 1.1 + return (66 + w_const[gender] * weight + 12.7 * height - 6.76 * age) * 1.1 if __name__ == '__main__':</pre>

Voltar para task 4 (4.1)

Command

```
git log --oneline -p -1
```

Expected output

```
32ef741 kilos improvements
diff --git a/fit.py b/fit.py
index a7e4384..33865f7 100755
--- a/fit.py
+++ b/fit.py
@@ -11,7 +11,9 @@ def main():
     parser.add_argument('age', type=int)
     parser.add_argument('weight', type=int, help='in pounds')
     parser.add_argument('height', type=int, help='in inches')
-    parser.add_argument('-k', '--kilos', action='store_true')
+    parser.add_argument(
+        '-k', '--kilos', action='store_true',
+        help='if true, then the weight given as input is interpreted as kg')
     args = parser.parse_args()

     result = rmb(
@@ -21,12 +23,14 @@ def main():

     def rmb(gender, age, weight, height, use_kg=False):
-        w_const_female = 9.56 if use_kg else 4.35
-        w_const_male = 13.75 if use_kg else 6.25
+        w_const = {
+            'female': 9.56 if use_kg else 4.35,
+            'male': 13.75 if use_kg else 6.25
+        }
         if gender == 'female':
-            return (655 + w_const_female * weight + 4.7 * height - 4.7 * age) * 1.1
+            return (655 + w_const[gender] * weight + 4.7 * height - 4.7 * age) * 1.1
         else:
-            return (66 + w_const_male * weight + 12.7 * height - 6.76 * age) * 1.1
+            return (66 + w_const[gender] * weight + 12.7 * height - 6.76 * age) * 1.1

     if __name__ == '__main__':
```

```
git log --oneline
```

```
32ef741 kilos improvements
c34c636 kilos feature
c23a513 added readme
333ed80 first cut at fit-cli app
85e982f Initialize repository
```

Voltar para task 4 (4.2)

Command	Expected output
git branch	* feat/meters feat/kilos master
git diff	diff --git a/fit.py b/fit.py index c28ba26..6af4580 100755 --- a/fit.py +++ b/fit.py @@ -11,14 +11,22 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') + parser.add_argument(+ '-m', '--meters', action='store_true', + help='if true, the height input will be interpreted in meters') args = parser.parse_args() - result = rmb(args.gender, args.age, args.weight, args.height) + result = rmb(+ args.gender, args.age, args.weight, args.height, use_m=args.meters) print('Your resting metabolic rate is {}'.format(result)) -def rmb(gender, age, weight, height): +def rmb(gender, age, weight, height, use_m=False): + h_const= { + 'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7 + } if gender == 'female': return (655 + 4.35 * weight + 4.7 * height - 4.7 * age) * 1.1 else:

Voltar para task 4 (4.2)

1) Atualize a função rmb para:

```
def rmb(gender, age, weight, height, use_m=False):
    w_const = {
        'female': 1.8 if use_m else 4.7,
        'male': 5 if use_m else 12.7
    }
    if gender == 'female':
        return (655 + 4.35 * weight + h_const[gender] * height - 4.7 * age) * 1.1
    else:
        return (66 + 6.25 * weight + h_const[gender] * height - 6.76 * age) * 1.1
```


Voltar para task 4 (4.1)

Command	Expected output
<pre>git diff</pre>	<pre>diff --git a/fit.py b/fit.py index c28ba26..2f41016 100755 --- a/fit.py +++ b/fit.py @@ -11,18 +11,26 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') + parser.add_argument(+ '-m', '--meters', action='store_true', + help='if true, the height input will be interpreted in meters') args = parser.parse_args() - result = rmb(args.gender, args.age, args.weight, args.height) + result = rmb(+ args.gender, args.age, args.weight, args.height, use_m=args.meters) print('Your resting metabolic rate is {}'.format(result)) -def rmb(gender, age, weight, height): +def rmb(gender, age, weight, height, use_m=False): + h_const = { + 'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7 + } + if gender == 'female': + return (655 + 4.35 * weight + 4.7 * height - 4.7 * age) * 1.1 + return (655 + 4.35 * weight + h_const[gender] * height - 4.7 * age) * 1.1 + else: + return (66 + 6.25 * weight + 12.7 * height - 6.76 * age) * 1.1 + return (66 + 6.25 * weight + h_const[gender] * height - 6.76 * age) * 1.1 if __name__ == '__main__':</pre>

Voltar para task 4 (4.1)

Command	Expected output		
git log --oneline -p -1	<pre>1495085 meters feature diff --git a/fit.py b/fit.py index c28ba26..2f41016 100755 --- a/fit.py +++ b/fit.py @@ -11,18 +11,26 @@ def main(): parser.add_argument('age', type=int) parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') + parser.add_argument(+ '-m', '--meters', action='store_true', + help='if true, the height input will be interpreted in meters') args = parser.parse_args() - result = rmb(args.gender, args.age, args.weight, args.height) + result = rmb(+ args.gender, args.age, args.weight, args.height, use_m=args.meters) print('Your resting metabolic rate is {}'.format(result)) -def rmb(gender, age, weight, height): +def rmb(gender, age, weight, height, use_m=False): + h_const = { + 'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7 + } + if gender == 'female': + return (655 + 4.35 * weight + 4.7 * height - 4.7 * age) * 1.1 + return (655 + 4.35 * weight + h_const[gender] * height - 4.7 * age) * 1.1 + else: + return (66 + 6.25 * weight + 12.7 * height - 6.76 * age) * 1.1 + return (66 + 6.25 * weight + h_const[gender] * height - 6.76 * age) * 1.1 if __name__ == '__main__':</pre>	git log --oneline	<pre>1495085 meters feature c23a513 added readme 333ed80 first cut at fit-cli app 85e982f Initialize repository</pre>

Material para Task 5

Voltar para task 5 (5.1)

Command	Expected output
<code>git log --oneline master</code>	<code>f79c92b Merge branch 'feat/kilos' into tmp 32ef741 kilos improvements 9a33c59 fix error checking bug c23a513 added readme c34c636 kilos feature 333ed80 first cut at fit-cli app 85e982f Initialize repository</code>

Voltar para task 5 (5.1)

Command	Expected output
git status	rebase in progress; onto db5bab You are currently rebasing branch 'feat/meters' on 'db5bab'. (fix conflicts and then run "git rebase --continue") (use "git rebase --skip" to skip this patch) (use "git rebase --abort" to check out the original branch)
git diff	Unmerged paths: (use "git reset HEAD <file>..." to unstage) (use "git add <file>..." to mark resolution)
	both modified: fit.py
	no changes added to commit (use "git add" and/or "git commit -a")

```

git diff
(cont)

++<<<<<< HEAD
+def rmb(gender, age, weight, height, use_kg=False):
+    w_const = {
+        'female': 9.56 if use_kg else 4.35,
+        'male': 13.75 if use_kg else 6.25
+    }
+    if gender == 'female':
+        return (655 + w_const[gender] * weight + 4.7 * height - 4.7 * age) +
1.1
+    else:
+        return (66 + w_const[gender] * weight + 12.7 * height - 6.76 * age)
+ 1.1
++++=====
+def rmb(gender, age, weight, height, use_m=False):
+    h_const= {
+        'female': 1.8 if use_m else 4.7,
+        'male': 5 if use_m else 12.7
+    }
+    if gender == 'female':
+        return (655 + 4.35 * weight + h_const[gender] * height - 4.7 * age)
+ 1.1
+    else:
+        return (66 + 6.25 * weight + h_const[gender] * height - 6.76 * age)
+ 1.1
++++>>>>>> meters feature

if __name__ == '__main__':

```

Voltar para task 5 (5.1)

1) Atualize a função main para:

```
def main():
    parser = argparse.ArgumentParser(
        description=(
            'fit-cli: compute your resting metabolic rate from the command line'))
    parser.add_argument('gender', choices=['female', 'male'])
    parser.add_argument('age', type=int)
    parser.add_argument('weight', type=int, help='in pounds')
    parser.add_argument('height', type=int, help='in inches')
    parser.add_argument(
        '-k', '--kilos', action='store_true',
        help='if true, then the weight given as input is interpreted as kg')
    parser.add_argument(
        '-m', '--meters', action='store_true',
        help='if true, the height input will be interpreted in meters')
    args = parser.parse_args()

    result = rmb(
        args.gender, args.age, args.weight, args.height,
        use_kg=args.kilos, use_m=args.meters)

    print('Your resting metabolic rate is {0}'.format(result))
```

Voltar para task 4 (4.1)

Command	Expected output
git diff	<pre>diff --cc fit.py index 34c5706,2f41016..0000000 --- a/fit.py +++ b/fit.py @@@ -12,25 -12,25 +12,41 @@@ def main() parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') parser.add_argument(+ '-k', '--kilos', action='store_true', + help='if true, then the weight given as input is interpreted as kg') ++ parser.add_argument(+ '-m', '--meters', action='store_true', + help='if true, the height input will be interpreted in meters') args = parser.parse_args() result = rmb(- args.gender, args.age, args.weight, args.height, use_kg=args.kilos) - args.gender, args.age, args.weight, args.height, use_m=args.meters) ++ args.gender, args.age, args.weight, args.height, ++ use_kg=args.kilos, use_m=args.meters) print('Your resting metabolic rate is {}'.format(result)) ++<<<<<<< HEAD ++def rmb(gender, age, weight, height, use_kg=False): ++ w_const = { ++ 'female': 9.56 if use_kg else 4.35, ++ 'male': 13.75 if use_kg else 6.25 ++ } ++ if gender == 'female': ++ return (655 + w_const[gender] * weight + 4.7 * height - 4.7 * age) * 1.1 ++ else: ++ return (66 + w_const[gender] * weight + 12.7 * height - 6.76 * age) * 1.1</pre>
git diff (cont)	<pre>+++++++ ++ def rmb(gender, age, weight, height, use_m=False): ++ h_const= { ++ 'female': 1.8 if use_m else 4.7, ++ 'male': 5 if use_m else 12.7 ++ } ++ if gender == 'female': ++ return (655 + 4.35 * weight + h_const[gender] * height - 4.7 * age) * 1.1 ++ else: ++ return (66 + 6.25 * weight + h_const[gender] * height - 6.76 * age) * 1.1 ++++>>>>>> meters feature if __name__ == '__main__':</pre>

Voltar para task 5 (5.1)

Command	Expected output
<code>git branch</code>	<code>feat/kilos</code> <code>feat/meters</code> <code>* master</code>

Voltar para task 5 (5.2)

1) Substitua:

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

Por:

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

Voltar para task 5 (5.2)

Command	Expected output
git diff	<pre>diff --git a/fit.py b/fit.py index 34c5706..27d1599 100755 --- a/fit.py +++ b/fit.py @@ -1,4 +1,4 @@ -#!/usr/bin/python +#!/usr/bin/env python import argparse</pre>

Voltar para task 5 (5.2)

Command	Expected output
<code>git log --oneline -p -1</code>	<pre>a9a2895 fix shebang diff --git a/fit.py b/fit.py index 34c5706..27d1599 100755 --- a/fit.py +++ b/fit.py @@ -1,4 +1,4 @@ -#!/usr/bin/python +#!/usr/bin/env python import argparse</pre>
<code>git log --oneline</code>	<pre>a9a2895 fix shebang f79c92b Merge branch 'feat/kilos' into tmp 32ef741 kilos improvements 9a33c59 fix error checking bug c23a513 added readme c34c636 kilos feature 333ed80 first cut at fit-cli app 85e982f Initialize repository</pre>

Voltar para task 5 (5.3)

Command	Expected output
git status	<pre>rebase in progress; onto 0910678 You are currently rebasing branch 'feat/meters' on '0910678'. (fix conflicts and then run "git rebase --continue") (use "git rebase --skip" to skip this patch) (use "git rebase --abort" to check out the original branch) Unmerged paths: (use "git reset HEAD <file>..." to unstage) (use "git add <file>..." to mark resolution) both modified: fit.py no changes added to commit (use "git add" and/or "git commit -a")</pre>
git diff	<pre>diff --cc fit.py index 27d1599,2f41016..0000000 --- a/fit.py +++ b/fit.py @@ -12,25 -12,25 +12,41 @@ def main() parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') parser.add_argument(+ '-k', '--kilos', action='store_true', + help='if true, then the weight given as input is interpreted as kg') + parser.add_argument(+ '-m', '--meters', action='store_true', + help='if true, the height input will be interpreted in meters') args = parser.parse_args() result = rmb(- args.gender, args.age, args.weight, args.height, use_kg=args.kilos) - args.gender, args.age, args.weight, args.height, use_m=args.meters) + args.gender, args.age, args.weight, args.height, + use_kg=args.kilos, use_m=args.meters) print('Your resting metabolic rate is {}'.format(result))</pre> <pre>git diff (cont) ++<<<<<< HEAD +def rmb(gender, age, weight, height, use_kg=False): + w_const = { + 'female': 9.56 if use_kg else 4.35, + 'male': 13.75 if use_kg else 6.25 + } + if gender == 'female': + return (655 + w_const[gender] * weight + 4.7 * height - 4.7 * age) * 1.1 + else: + return (66 + w_const[gender] * weight + 12.7 * height - 6.76 * age) * 1.1 +===== +def rmb(gender, age, weight, height, use_m=False): + h_const = { + 'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7 + } + if gender == 'female': + return (655 + 4.35 * weight + h_const[gender] * height - 4.7 * age) * 1.1 + else: + return (66 + 6.25 * weight + h_const[gender] * height - 6.76 * age) * 1.1 +>>>>>>> meters feature if __name__ == '__main__':</pre>

Voltar para task 5 (5.3)

1) Atualize a função rmb para:

```
def rmb(gender, age, weight, height, use_kg=False, use_m=False):
    w_const = {
        'female': 9.56 if use_kg else 4.35,
        'male': 13.75 if use_kg else 6.25
    }
    h_const = {
        'female': 1.8 if use_m else 4.7,
        'male': 5 if use_m else 12.7
    }
    if gender == 'female':
        return (
            655 + w_const[gender] * weight + h_const[gender] * height -
            4.7 * age) * 1.1
    else:
        return (
            66 + w_const[gender] * weight + h_const[gender] * height -
            6.76 * age) * 1.1
```

Voltar para task 5 (5.3)

Com mand	Expected output
git diff	<pre>diff --cc fit.py index 27d1599,2f41016..0000000 --- a/fit.py +++ b/fit.py @@@ -12,25 -12,25 +12,37 @@@ def main() parser.add_argument('weight', type=int, help='in pounds') parser.add_argument('height', type=int, help='in inches') parser.add_argument(+ '-k', '--kilos', action='store_true', + help='if true, then the weight given as input is interpreted as kg') ++ parser.add_argument(+ '-m', '--meters', action='store_true', + help='if true, the height input will be interpreted in meters') + args = parser.parse_args() result = rmb(- args.gender, args.age, args.weight, args.height, use_kg=args.kilos) - args.gender, args.age, args.weight, args.height, use_m=args.meters) ++ args.gender, args.age, args.weight, args.height, ++ use_kg=args.kilos, use_m=args.meters) print('Your resting metabolic rate is {}'.format(result))</pre>
git diff (cont)	<pre>- def rmb(gender, age, weight, height, use_kg=False): -def rmb(gender, age, weight, height, use_m=False): ++def rmb(gender, age, weight, height, use_kg=False, use_m=False): + w_const = { + 'female': 9.56 if use_kg else 4.35, + 'male': 13.75 if use_kg else 6.25 + } + h_const= { + 'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7 + } + if gender == 'female': - return (655 + w_const[gender] * weight + 4.7 * height - 4.7 * age) * 1.1 - return (655 + 4.35 * weight + h_const[gender] * height - 4.7 * age) * 1.1 ++ return (++ 655 + w_const[gender] * weight + h_const[gender] * height - ++ 4.7 * age) * 1.1 + else: - return (66 + w_const[gender] * weight + 12.7 * height - 6.76 * age) * 1.1 - return (66 + 6.25 * weight + h_const[gender] * height - 6.76 * age) * 1.1 ++ return (++ 66 + w_const[gender] * weight + h_const[gender] * height - ++ 6.76 * age) * 1.1 if __name__ == '__main__':</pre>

Voltar para task 5 (5.3)

Command	Expected output	
<pre>git log --oneline -p -1</pre>	<pre>b530b05 meters feature diff --git a/fit.py b/fit.py index 27d1599..e95c109 100755 --- a/fit.py +++ b/fit.py @@ -14,23 +14,35 @@ def main(): parser.add_argument('-k', '--kilos', action='store_true', help='if true, then the weight given as input is interpreted as kg') + parser.add_argument(+ '-m', '--meters', action='store_true', + help='if true, the height input will be interpreted in meters') args = parser.parse_args() result = rmb(- args.gender, args.age, args.weight, args.height, use_kg=args.kilos) + args.gender, args.age, args.weight, args.height, + use_kg=args.kilos, use_m=args.meters) print('Your resting metabolic rate is {}'.format(result))</pre>	<pre>git log -- oneline -p -1 (cont) -def rmb(gender, age, weight, height, use_kg=False): +def rmb(gender, age, weight, height, use_kg=False, use_m=False): -- w_const = { 'female': 9.56 if use_kg else 4.35, 'male': 13.75 if use_kg else 6.25 } + h_const = { + 'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7 + } if gender == 'female': - return (655 + w_const[gender] * weight + 4.7 * height - 4.7 * age) * 1.1 + return (+ 655 + w_const[gender] * weight + h_const[gender] * height - + 4.7 * age) * 1.1 else: - return (66 + w_const[gender] * weight + 12.7 * height - 6.76 * age) * 1.1 + return (+ 66 + w_const[gender] * weight + h_const[gender] * height - + 6.76 * age) * 1.1 if __name__ == '__main__':</pre>
		<pre>git log -- oneline 2cad7a meters feature 0910678 fix shebang dbe5bab Merge branch 'feat/kilos' into tmp c8aba68 kilos improvements f0fc9a7 fix error checking bug bbc9a04 kilos feature 216a831 added readme 0e032fd first cut at fit-cli app f821dce Initialize repository</pre>

Material para Task 6

Voltar para task 6 (6.1)

Command	Expected output
<code>git branch</code>	<code>feat/kilos</code> <code>feat/meters</code> <code>* master</code>
<code>git log --oneline</code>	<code>01ef275 meters feature</code> <code>2d5f2ca fix shebang</code> <code>c03ce28 Merge branch 'feat/kilos' into tmp</code> <code>238a883 kilos improvements</code> <code>0865880 fix error checking bug</code> <code>b8284fc kilos feature</code> <code>2b2442b added readme</code> <code>6eb27d8 first cut at fit-cli app</code> <code>7525920 Initialize repository</code>

Voltar para task 6 (6.1)

1) Atualize a função rmb para:

```
def rmb(gender, age, weight, height, use_kg=False, use_m=False):
    const = {
        'weight': {
            'female': 9.56 if use_kg else 4.35,
            'male': 13.75 if use_kg else 6.25
        },
        'height': {
            'female': 1.8 if use_m else 4.7,
            'male': 5 if use_m else 12.7
        }
    }
    if gender == 'female':
        return (
            655 + const['weight'][gender] * weight +
            const['height'][gender] * height - 4.7 * age) * 1.1
    else:
        return (
            66 + const['weight'][gender] * weight +
            const['height'][gender] * height - 6.76 * age) * 1.1
```

Voltar para task 6 (6.1)

Command	Expected output
<code>git diff</code>	<pre>diff --git a/fit.py b/fit.py index e95c109..a080817 100755 --- a/fit.py +++ b/fit.py @@ -27,22 +27,24 @@ def main(): def rmb(gender, age, weight, height, use_kg=False, use_m=False): w_const = { 'female': 9.56 if use_kg else 4.35, 'male': 13.75 if use_kg else 6.25 } h_const = { 'female': 1.8 if use_m else 4.7, 'male': 5 if use_m else 12.7 } const = { 'weight': { 'female': 9.56 if use_kg else 4.35, 'male': 13.75 if use_kg else 6.25 }, 'height': { 'female': 1.8 if use_m else 4.7, 'male': 5 if use_m else 12.7 } } if gender == 'female': return (655 + w_const[gender] * weight + h_const[gender] * height - 4.7 * age) * 1.1 + 655 + const['weight'][gender] * weight + + const['height'][gender] * height - 4.7 * age) * 1.1 else: return (66 + w_const[gender] * weight + h_const[gender] * height - 6.76 * age) * 1.1 + 66 + const['weight'][gender] * weight + + const['height'][gender] * height - 6.76 * age) * 1.1 if __name__ == '__main__':</pre>

Voltar para task 6 (6.1)

Command	Expected output
<pre>git log --oneline -p -1</pre>	<pre>7ce46e5 join const dicts diff --git a/fit.py b/fit.py index e95c109..a080817 100755 --- a/fit.py +++ b/fit.py @@ -27,22 +27,24 @@ def main(): def rmb(gender, age, weight, height, use_kg=False, use_m=False): - w_const = { - 'female': 9.56 if use_kg else 4.35, - 'male': 13.75 if use_kg else 6.25 - } - h_const= { - 'female': 1.8 if use_m else 4.7, - 'male': 5 if use_m else 12.7 + const = { + 'weight': { + 'female': 9.56 if use_kg else 4.35, + 'male': 13.75 if use_kg else 6.25 + }, + 'height': { + 'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7 + } + }</pre>
<pre>git log --oneline -p -1</pre>	<pre>if gender == 'female': return (655 + w_const[gender] * weight + h_const[gender] * height - 4.7 * age) * 1.1 + 655 + const['weight'][gender] * weight + + const['height'][gender] * height - 4.7 * age) * 1.1 else: return (66 + w_const[gender] * weight + h_const[gender] * height - 6.76 * age) * 1.1 + 66 + const['weight'][gender] * weight + + const['height'][gender] * height - 6.76 * age) * 1.1</pre>
<pre>git log --oneline</pre>	<pre>734ed30 join const dicts 2cad7a meters feature 0910678 fix shebang dbe5bab Merge branch 'feat/kilos' into tmp c8aba68 kilos improvements f0fc9a7 fix error checking bug bbc9a04 kilos feature 216a831 added readme 0e032fd first cut at fit-cli app f821dce Initialize repository</pre>

Voltar para task 6 (6.2)

Command	Expected output
<code>git status</code>	On branch master Your branch is up-to-date with 'origin/master'. nothing to commit, working directory clean

Voltar para task 6 (6.2)

- 1) Depois da linha 3 (depois do import), adicione o seguinte texto em uma nova linha:

```
import collections
```

- 2) Atualize a rmb para:

```
def rmb(gender, age, weight, height, use_kg=False, use_m=False):
    Const = collections.namedtuple('Const', ['weight', 'height'])
    const = Const(
        {'female': 9.56 if use_kg else 4.35,
         'male': 13.75 if use_kg else 6.25},
        {'female': 1.8 if use_m else 4.7,
         'male': 5 if use_m else 12.7})
    if gender == 'female':
        return (
            655 + const.weight[gender] * weight +
            const.height[gender] * height - 4.7 * age) * 1.1
    else:
        return (
            66 + const.weight[gender] * weight +
            const.height[gender] * height - 6.76 * age) * 1.1
```

Voltar para task 6 (6.2)

Command	Expected output
git diff	<pre>diff --git a/fit.py b/fit.py index e95c109...9467829 100755 --- a/fit.py +++ b/fit.py @@ -1,6 +1,7 @@ #!/usr/bin/env python import argparse +import collections def main(): @@ -27,22 +28,20 @@ def main(): def rmb(gender, age, weight, height, use_kg=False, use_m=False): - w_const = { - 'female': 9.56 if use_kg else 4.35, - 'male': 13.75 if use_kg else 6.25 - } - h_const = { - 'female': 1.8 if use_m else 4.7, - 'male': 5 if use_m else 12.7 - } + Const = collections.namedtuple('Const', ['weight', 'height']) + const = Const(+ {'female': 9.56 if use_kg else 4.35, + 'male': 13.75 if use_kg else 6.25}, + {'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7}) if gender == 'female': return (- 655 + w_const[gender] * weight + h_const[gender] * height - - 4.7 * age) * 1.1 + 655 + const.weight[gender] * weight + + const.height[gender] * height - 4.7 * age) * 1.1 else: return (- 66 + w_const[gender] * weight + h_const[gender] * height - - 6.76 * age) * 1.1 + 66 + const.weight[gender] * weight + + const.height[gender] * height - 6.76 * age) * 1.1</pre>

Voltar para task 6 (6.1)

Command	Expected output
<pre>git log -- online -p -1</pre>	<pre>9afbe76 switch to using nt for consts diff --git a/fit.py b/fit.py index e95c109..9467829 100755 --- a/fit.py +++ b/fit.py @@ -1,6 +1,7 @@ #!/usr/bin/env python import argparse +import collections def main(): @@ -27,22 +28,20 @@ def main(): def rmb(gender, age, weight, height, use_kg=False, use_m=False): - w_const = { - 'female': 9.56 if use_kg else 4.35, - 'male': 13.75 if use_kg else 6.25 - } - h_const = { - 'female': 1.8 if use_m else 4.7, - 'male': 5 if use_m else 12.7 - } + Const = collections.namedtuple('Const', ['weight', 'height']) + const = Const(+ {'female': 9.56 if use_kg else 4.35, + 'male': 13.75 if use_kg else 6.25}, + {'female': 1.8 if use_m else 4.7, + 'male': 5 if use_m else 12.7})</pre>

```
git log --
online -p
-1

if gender == 'female':
    return (
        655 + w_const[gender] * weight + h_const[gender] * height -
        4.7 * age) * 1.1
+ 655 + const.weight[gender] * weight +
+ const.height[gender] * height - 4.7 * age) * 1.1
else:
    return (
        66 + w_const[gender] * weight + h_const[gender] * height -
        6.76 * age) * 1.1
+ 66 + const.weight[gender] * weight +
+ const.height[gender] * height - 6.76 * age) * 1.1

if __name__ == '__main__':
```

```
git log
--
online
9afbe76 switch to using nt for consts
b530b05 meters feature
a9a2895 fix shebang
f79c92b Merge branch 'feat/kilos' into tmp
32ef741 kilos improvements
9a33c59 fix error checking bug
c34c636 kilos feature
c23a513 added readme
333ed80 first cut at fit-cli app
85e982f Initialize repository
```