

Práctica 1

Un vistazo rápido de Git

Antes de comenzar esta práctica es necesario tener git instalado. Asegurate que tengas git instalado ejecutando el siguiente comando:

```
$ sudo apt-get install git
```

Obteniendo ayuda

Para cualquier comando de git podés usar el subcomando help. Por ejemplo:

```
$ git help status
```

nos va a mostrar las opciones del subcomando status.

Otro punto a tener en cuenta es la salida del subcomando **status**. Dicha salida es sumamente detallada respecto de las acciones que se pueden hacer. Prestale atención!

1. ¿Cuales son los 3 estados? ¿Qué significa cada uno?

2. Cloná el repositorio de materiales de la materia:

```
git clone https://github.com/TTPS-ruby/capacitacion-ruby-ttps.git
```

3. ¿Cuál es el hash del commit recién creado? **Ayuda:** mirá el subcomando **log**.

4. ¿Qué hacen los comandos **add** y **commit**?

5. ¿Qué hace el subcomando **push**?

6. ¿Qué hace el subcomando **pull**?

7. ¿Qué hace el subcomando **fetch**?

8. Creá un archivo de texto dentro del directorio que se creó al clonar el repositorio y verificá el estado con el subcomando **status**.

9. Eliminá los archivos `untracked` usando el subcomando **clean**.

10. Desplazate al commit cuyo hash empieza con `ad3252` para ver el primer commit de este repositorio:

```
git checkout ad3252
```

Cuando termines volvé a la última versión descargada con:

```
git checkout master
```

11. Para poder tener la versión más actualizada de los materiales ejecutá el subcomando **pull** periódicamente.

Ruby: Tipos básicos y sintaxis

Nota: No usar `while`, `for` ni `repeat`.

1. Instale ruby 2.1.2 usando [rbenv](#) y [ruby-build](#). En el primer link se encuentra la guía de instalación.

Primero es necesario instalar algunos paquetes (distribuciones basadas en Debian):

```
apt-get install autoconf bison build-essential libssl-dev libyaml-dev\
libreadline6 libreadline6-dev zlib1g zlib1g-dev libsqlite3-dev
```

Observación: En distribuciones basadas en Debian usar `~/.bashrc` en lugar de `~/.bash_profile`:

```
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
$ echo 'eval "$(rbenv init -)"' >> ~/.bashrc
```

Para otras distribuciones ver el readme de [rbenv](#) y la wiki de [ruby-build](#).

2. Investigue y pruebe como crear objetos de los siguientes tipos usando literales y usando **new** (cuando se pueda):

- Un arreglo.
- Un hash.
- Un string.
- Un símbolo.

3. Investigue que métodos se pueden usar para:

- Obtener la lista de métodos de una clase.
- Obtener la lista de métodos de instancia de un clase.
- Obtener la lista de variables de instancia de una clase.
- Obtener la lista de ancestros de una clase.

4. ¿Cuál es la diferencia entre las siguientes sentencias?:

- `String.methods`
- `'Hola'.methods`
- `String.methods false`

5. Escriba una función que reciba un **hash** y retorne un **string** con las claves y valores del **hash** formateados en una lista HTML.

Por ejemplo:

```
> to_html({bananas: 5, naranjas: 10})
=> "<ul><li>bananas: 5</li><li>naranjas: 10</li></ul>"
```

6. Dado un string cualquiera reemplace cada ocurrencia de `{` por `do\n` y cada ocurrencia de `}` por `end`. Por ejemplo:

```
> reemplazar("[1, 2, 3, 4].each { |x| a = x**2 + x**3 + x**5\np a\n }")
=> "[1, 2, 3, 4].each do\n |x| a = x**2 + x**3 + x**5\np a\n end"
```

¿Puede escribir la función reemplazar en una sola linea aprovechando los métodos de instancia de la clase **String**?

7. Si listamos todos los números naturales menores que 10 que son múltiplos de 3 o 5 obtenemos 3, 5, 6 y 9. La suma de todos estos números es 23. Encontrá la suma de todos los múltiplos de 3 o 5 menores que 1000.
8. Cada nuevo término en la secuencia de Fibonacci es generado sumando los 2 términos anteriores. Los primeros 10 términos son: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55. Considerando los términos en la secuencia de Fibonacci cuyos valores no exceden los 4 millones, encontrá la suma de los términos pares.
9. Un número palíndromo se lee igual al derecho y al revés. El número palíndromo más grande obtenido de la multiplicación de dos números de 2 dígitos es 9009 (91 x 99 = 9009). Encontrá el palíndromo más grande obtenido a través de la multiplicación de dos números de 3 dígitos.
10. 2520 es el número más chico que puede ser dividido por cada uno de los números del 1 al 10 sin obtener resto. ¿Cual es el número más chico que puede ser dividido por cada uno de los números del 1 al 20?

11. La suma de los cuadrados de los primeros 10 números naturales es 385 ($1^2 + 2^2 + \dots + 10^2 = 385$). El cuadrado de la suma de los primeros 10 números naturales es 3025 ($(1 + 2 + \dots + 10)^2 = 55^2 = 3025$). Por lo tanto, la diferencia entre el cuadrado de la suma y la suma de los cuadrados de los primeros 10 números naturales es 2640 ($3025 - 385 = 2640$). Encontrá la diferencia entre el cuadrado de la suma y la suma de los cuadrados de los primeros 100 números naturales.
12. La lista de los primeros 6 números primos es 2, 3, 5, 7, 11 y 13. Se puede observar que el 6to número primo es 13. ¿Cual es el número primo nro 10001?
13. La suma de los primos menores que 10 es 17 ($2 + 3 + 5 + 7 = 17$). Encontrá la suma de todos los primos menores que 2 millones.
14. Dado un arreglo de strings cualquiera, es necesario escribir un método que devuelva un arreglo con la longitud de dichos strings. Ejemplo: dado ['Ruby', 'is', 'awesome'] debe retornar [4, 2, 7]
15. Analizar y probar el siguiente código:

```
[ :upcase, :downcase, :capitalize, :swapcase ].map do |symbol|  
  "Hola Mundo".send(symbol)  
end
```

16. a. Dado un color expresado como una combinación RGB calcular su representación entera. Consideramos que un color rgb se expresa como un hash con las claves [:red, :green, :blue], y para cada una toma valores en el rango (0..255). Por ejemplo:

- { red: 0, green: 0, blue: 255 }
- { red: 128, green: 128, blue: 255 }

La representación entera se calcula como: $\text{red} + \text{green} * 256 + \text{blue} * 256^2$

b. Realizar el mismo cálculo obteniendo los coeficientes para cada componente del color de otro hash coefficients = { red: 256^{**0} , green: 256^{**1} , blue: 256^{**2} }