Basic-Ruby-Style-Guide

Semi colons ;

Avoid them unless writing multiple statements per line

Example:

```ruby
# Good
def needs_cursor?; true; end

# Bad
name = gets.chomp;
puts("hi #{name}");

# Even worse - uses inconsistently
name = gets.chomp
puts("hi #{name}");
```

Variables

Names

Must be in the snake_case format.

Example:

```ruby
#Good
name = "paul"
first_name = "paul"

#Bad
Name = "paul"
FamilyName = "Sarda"
Family_Name = "Sarda"
```

Operators

Must have a space on both sides of the operator always.

Example:

```
# Good
paul_is_cool = true

# Bad
paul_is_not_cool =false
paul_is_not_cool=false
paul_is_not_cool= false

# Good
ten = 5 + 5

# Bad
ten = 5+ 5
```

functions

Names

Function names should use snake_case.

Examples:

```
# Good
def return_paul()
    return "paul"
end

# Bad - PascalCase

def returnPaul()
    return "paul"
end

# Also Bad - camelCase
def ReturnPaul()
    return "paul"
end
```

Parentheses (round brackets)

Calls

Use parentheses whenever there is more one or more arguments. Or always use it.

Examples:

```ruby
# Good
puts("Hi my name is paul") # needed
name = gets().chomp() # not needed no arguments

puts("Hi my name is paul")
name = gets.chomp()

# Bad
# inconsistent
name = gets.chomp()
last_name = gets.chomp

# Missing parentheses
puts "Hi my name is paul"
```

No spaces between (and the first argument. No spaces between) and the last argument.

Examples:

```ruby
# Good
puts("paul is so cool")

# Bad
puts( "paul is not cool" )
```

Definitions

Add parentheses if there is one or more arguments.

Example:

```ruby
def sum(a, b) # needed
    return (a + b)
end

```

```
def return_paul # no brackets needed
    return "paul"
end
```

Return

The return keyword must always be used if a function returns a value.

Example:

```
# Good
 def return_paul()
    return "Paul"
end

# Good becuase the function doesn't need to return anything
def print_paul()
    puts("Paul")
end

# Bad
# missing return
def return_paul
    "paul"
end
```

If

Don’t use Parentheses or use them not both

(use is preferred) Example:

```
# Good
if name === "paul"
    puts("Woah you are so cool")
end

if course == "BA-CS"
    puts("Good choice")
end

```

```
    # Good
    if (name === "paul")
        puts("Woah you are so cool")
    end

    if (course == "BA-CS")
        puts("Good choice")
    end

    #Bad becuase inconsistent
    if (name === "paul")
        puts("Woah you are so cool")
    end

    if course == "BA-CS"
        puts("Good choice")
    end
```

Always have a space between if and the condition.

Example:

```
    # Good
    if (name === "paul")
        puts("Woah you are so cool")
    end

    # Bad
    if(name === "paul")
        puts("Woah you are so cool")
    end
```

Case

Each when should be the same level of indentation as case

Example:

```
    # Good
    case name
    when "paul"
        puts("Old name");
    when "andrew"
        puts("Okay name");
    when "john"
        puts("Great name");
    else
        puts("bad name");
    end

    # Bad
    case name
        when "paul"
            puts("Old name");
        when "andrew"
            puts("Okay name");
        when "john"
            puts("Great name");
        else
            puts("bad name");
    end

    case name
        when "paul"
        puts("Old name");
        when "andrew"
        puts("Okay name");
        when "john"
        puts("Great name");
        else
        puts("bad name");
    end
```

Loops

While

Same condition rules as if.

Example:

```
    while (i <= 50)  do
        puts("value of i is #{i}");
        i += 1;
    end
```

For

Example:

```
    for i in 1..50 do
        puts("value of i is #{i}");
    end
```

Array Example:

```
    dice_rolls = [1, 3, 3, 1];

    for roll in dice_rolls do
        puts("#{roll}");
    end
```

Indentation

Everything between the end and what the end is ending must be indented

Examples:

```
    # Good
    def its_paul()
        puts("REALLY?")
    end

    def main()
        name = "paul"

        if name === "paul"
            its_paul()
        end
    end

    # BAD
            def its_paul
```

```
puts("REALLY?")
end

    def main
name = "paul"

if name ===             "paul"
                    its_paul
            end
end

# Also BAD

def its_paul
puts("REALLY?")
end

def main
name = "paul"

if name === "paul"
its_paul
end
end

# Also BAD

def its_paul
        puts("REALLY?")
end

def main
    name = "paul"

    if name === "paul"
            its_paul
    end
end
```

Structures / Records / Classes

Name

Must be in PascalCase meaning that every word should be capitalised and no spaces.

Members must be in snake_case

Example:

```ruby
class Person
    attr_accessor :name, :birth_year
end

class EvilTeacher
    attr_accessor :subject, :year_started
end
```

Array

All normal naming rules apply

Example:

```ruby
numbers = [];

numbers << 10;

puts(numbers[0]);
```