Arrgh!

A programming Language by  
Alan Renato Bunese

Todas as instruções que utilizam-se **desta cor** são palavras-chaves:

* **Piracy** – Define o começo do programa, deve ser seguido pelo nome do programa. Deve ser sempre seguido por um **Ahoy!** e um **Heave To**.
* **Ahoy!** – Define o começo de um bloco de Código, pode ser um programa, uma função ou até um condicional.
* **Heave To** – Define o final de um bloco de Código, sempre deve ser utilizado depois de um **Ahoy!**
* **Be** – Define uma variável, seguida do nome e do tipo. Exemplo: **Be** Name: Pieces of Eight
* **Plunder** – Define uma função, deve ser sempre seguida do tipo de retorno, nome e os inputs entre parênteses com os tipos e um **Ahoy!** e um **Heave To** que define sua funcionalidade.
* **Booty** – Retorna um valor de uma função.
* **What** – Inicia um bloco condicional. Deve ser seguido por uma condição que retorne Aye ou Nay.
* **On Top Of** – Operação de divisão. Exemplo: 10 **On Top Of** 20 = 10/20 = 0.5
* **Times** – Operação de multiplicação. Exemplo: 10 **Times** 20 = 10\*20 = 200
* **With** – Operação de adição. Exemplo: 10 **With** 20 = 10 + 20 = 30
* **Without** – Operação de subtração. Exemplo: 10 **Without** 20 = 10 – 20 = -10
* **Is Equals To** – Operação condicional, precendido por um Bounty e sucedido por outro Bounty. Vertifica se A = B.
* **Is Greater Then** – Operação condicional, precendido por um Bounty e sucedido por outro Bounty. Realiza a operação a seguir: A > B.
* **Is Less Then** – Operação condicional, precendido por um Bounty e sucedido por outro Bounty. Realiza a operação a seguir: A < B.
* **Or** – Operação condicional, precendido por outra operação condicional ou um Aye ou Nay e sucedido por outra operação condicional ou um Aye ou Nay. Retorna Aye caso uns dos valores sejam Aye.
* **Scuttle** – Deve ser precendido por um bloco **Ahoy!**, apenas depois de um **What**. O bloco contido dentro desta palavra-chave é executado caso as condições do **What** retornem Nay.

Todos os comentários são feitos de um jeito, apenas: Belay that Order: seguido do comentário que você queira.

Todas as instruções que utlizam-se desta cor são tipos:

* Pieces of Eight – Define um caractere.
* Multiple Pieces of Eight – Define uma string. Caso haja um caractere de escape %, seguido de uma variável, o valor deste variável é adicionado ao output final.
* Bounty – Define um número com decimais.

Chamada de funções são feitas assim: NomeDaFunção(Parâmetro1, Parâmetro2, …). Aqui está umas funções:

* Walk the plank – Recebe um parâmetro em formato Multiple Pieces of Eight. A função escreve no output a mensagem do parâmetro.
* Feed the fish – Recebe um parâmetro em formato Multiple Pieces of Eight. A função escreve no output a mensagem do parâmetro e espera o retorno do usuário. Caso seja atribuido a uma variável, o valor é convertido ao tipo desejado, se possível.

Belay that Order: Define our Program.

**Piracy** CalculatePresence

**Ahoy!**

Belay that Order: Define the amount of classes, given as an input.

**Be** Classes**:** Bounty **:=** Feed the fish(“Insert the Number of Classes:”)

Belay that Order: Print out the amount of classes.

Walk the plank(“The amount of classes is %Classes.”)

Belay that Order: Define the amount of classes the student went to.

**Be** WentTo**:** Bounty **:=** Feed the fish(“Insert the Amount of Classes the Student went to:”)

Belay that Order: Define the Presence Percentage Function.

**Plunder** Bounty GetPresencePercentage(Classes**:** Bounty, WentTo**:** Bounty)

**Ahoy!**

**Booty** WentTo **On Top Of** Classes

**Heave To**

Belay that Order: Define the percentage of presence.

**Be** Percentage**:** Bounty **:=** GetPresencePercentage (Bounty, WentTo) **Times** 100

Belay that Order: If our percentage is greater then 75, this student was approved.

**What** Percentage **is Greater Then** 75

**Ahoy!**

Walk the plank(“Student approved with %Percentage%% of presence.”)

**Scuttle**

Walk the plank(“Student was not approved with %Percentage%% of presence.”)

**Heave To**

**Heave To**

# Código Em Python

# Define the amount of classes, given as an input.

Classes = int(input(“Insert the Number of Classes:”))

# Print the amount of classes.

print(“The amount of classes is “ + str(Classes))

# Define the amount of classes the student went to.

WentTo = int(input(“Insert the Number of Classes the student went to:”))

# Define the presence Percentage function.

**def** GetPresensePercentage(Classes, WentTo):

**return** WentTo **/** Classes

# Define the percentage of presence.

Percentage = GetPresencePercentage(Classes, WentTo) **\*** 100

# If our percentage is greater then 75, this student was approved.

**If** Percentage >= 75**:**

print(“Student was approved with “ + str(Percentage) + “% of presence.”)

**else:**

print(“Student was not approved with “ + str(Percentage) + “% of presence.”)