

Atividades

1. Faça um programa que informe a versão do Python que você está utilizando.

```
import sys
print(sys.version)
```

- **import sys:** é um módulo padrão do python
- **sys.version:** informa o número da versão do interpretador Python mais informações adicionais sobre o número da compilação e o compilador usado.

```
import platform
print(platform.python_version())
print(platform.system())
```

- **import platform:** este módulo recupera o máximo possível de dados de identificação de plataforma possível.
 - **platform.python_version():** Retorna a versão do Python como string.
 - **platform.system():** Retorna o nome do sistema.
-

2. Crie um algoritmo que receba um número, conte o número total de dígitos e mostre o resultado.

Por exemplo, se o número é 2021 , então a saída deve ser 4.

```
import math
numero = abs(int(input("Digite um numero: ")))
digitos = int(math.log(numero, 10) + 1)
print(f"O número tem {digitos} digitos.")
```

- **math:** módulo que fornece as funções matemáticas.
- **numero:** nome dado a variável.
- **abs:** função utilizada para pegar o valor absoluto do número que será digitado. Mesmo se o número for negativo, contanto que não seja complexo ou flutuante.
- **int:** método que retorna um número por inteiro.
- **input:** função que permite a entrada do usuário.
- **math.log:** retorna o logaritmo. de base 10.
- **f:** formatação de string.

3. Qual é o endereço de rede e de broadcast do IP válido 200.67.67.43/27?

R: O endereço de rede é: [200.67.67.32](#)

Justificativa:

De acordo com a representação binária de 27 bits

11111111.11111111.11111111.111 00000, me resta **5 hosts**, a qual eu preciso reservar **1** para o **domínio de broadcast**, restando portanto, **4 hosts**, que elevado a potência de 2, resulta em 32 bits.

R: O endereço de broadcast é: [200.67.67.63](#)

Justificativa:

Ao pegar a máscara de rede [255.255.255.224](#) e transformar em binário tenho a seguinte representação: **11111111.11111111.11111111.111 00000**.

Para saber o endereço de broadcast, preciso converter a representação binária da máscara de rede para wild card, ou seja onde tenho bits de nº 1, converto para zero e onde tenho bits zero, converto em bits de nº 1. Logo a representação do wild card fica:

11111111.11111111.11111111.111 00000 ==> máscara de rede

00000000.00000000.00000000.000 11111 ==> wild card

Apesar de ter 5 bits de nº 1 no wild card, reserva-se 1 para domínio de rede, restando então 4 bits de nº1, elevados a potência de 2. Logo tenho o resultado de nº 63. Pois $2^4 - 1 = 15$.

4. Dado o IP 192.168.10.5 e a Máscara de sub-rede 255.255.255.192 determine quantas sub-redes podem ser criadas e as faixas de cada sub-rede criada.

R: 4 sub-redes podem ser criadas.

Faixas de sub-rede.

[192.168.10.1](#) - [192.168.10.62](#)

[192.168.10.65](#) - [192.168.10.126](#)

[192.168.10.129](#) - [192.168.10.190](#)

[192.168.10.193](#) - [192.168.10.254](#)

Cálculo:

$256 - 192 = 64$ IP's totais. 62 IP's reservando 1 para rede e 1 para broadcast.

$256/2 = 4$ sub-redes.

prefixo /26 (2 bits emprestados)

5. Quantas sub-redes podem ser criadas se pegarmos emprestados 04 bits para uma rede
IP Classe C.

R:16 sub-redes

Cálculo

$256-240 = 16$

4 bits

$128+64+32+16 = 240$

máscara de sub-rede: 255.255.255.240

prefixo /28

$24 \text{ bits} + 4 \text{ bits} = 28$

6 . Crie um playbook que execute em dois nodes a instalação do apache2 e a configuração de um index.html qualquer.

```
Vagrant.configure("2") do |master|
  master.vm.define "master" do |master|
    master.vm.box = "ubuntu/jammy64"
    master.vm.hostname = "master"
    master.vm.network "public_network", bridge: "Realtek 8811CU Wireless LAN 802.11ac USB
NIC", ip: "192.168.0.42"
    master.vm.boot_timeout = 1200
    master.vm.synced_folder "C:/Users/jessi/LABSVAGRANT/Master", "/home/ansible/playbook/"
    master.vm.provision "shell", inline: "sudo apt update -y && sudo apt dist-upgrade -y"
    master.vm.provision "shell", inline: "sudo apt-get install python-is-python3 -y"
    master.vm.provision "shell", inline: "sudo apt-get install python3-pip -y"
    master.vm.provision "shell", inline: "sudo python3 -m pip install --upgrade pip"
    master.vm.provision "shell", inline: "sudo pip install ansible==2.9.10"
    master.vm.provider "virtualbox" do |master|
      master.memory = 1024
      master.cpus = 2
      master.name = "master"
    end
  end
end
```

```

Vagrant.configure("2") do |ubuntu2|
  ubuntu2.vm.define "ubuntu2" do |ubuntu2|
    ubuntu2.vm.box = "ubuntu/jammy64"
    ubuntu2.vm.hostname = "ubuntu2"
    ubuntu2.vm.network "public_network", bridge: "Realtek 8811CU Wireless LAN 802.11ac
USB NIC", ip: "192.168.0.43"
    ubuntu2.vm.boot_timeout = 1200
    ubuntu2.vm.provision "shell", inline: "sudo apt update -y && sudo apt dist-upgrade -
y"
    ubuntu2.vm.provision "shell", inline: "sudo apt-get install python-is-python3 -y"
    ubuntu2.vm.provision "shell", inline: "sudo apt-get install python3-pip -y"
    ubuntu2.vm.provision "shell", inline: "sudo python3 -m pip install --upgrade pip"
    ubuntu2.vm.provider "virtualbox" do |ubuntu2|
      ubuntu2.memory = 700
      ubuntu2.cpus = 2
      ubuntu2.name = "ubuntu2"
    end
  end
end
end

```

```

Vagrant.configure("2") do |ubuntu3|
  ubuntu3.vm.define "ubuntu3" do |ubuntu3|
    ubuntu3.vm.box = "ubuntu/jammy64"
    ubuntu3.vm.hostname = "ubuntu3"
    ubuntu3.vm.network "public_network", bridge: "Realtek 8811CU Wireless LAN 802.11ac
USB NIC", ip: "192.168.0.44"
    ubuntu3.vm.provision "shell", inline: "sudo apt update -y && sudo apt dist-upgrade -
y"
    ubuntu3.vm.provision "shell", inline: "sudo apt-get install python-is-python3 -y"
    ubuntu3.vm.provision "shell", inline: "sudo apt-get install python3-pip -y"
    ubuntu3.vm.provision "shell", inline: "sudo python3 -m pip install --upgrade pip"
    ubuntu3.vm.provider "virtualbox" do |ubuntu3|
      ubuntu3.memory = 512
      ubuntu3.cpus = 2
      ubuntu3.name = "ubuntu3"
    end
  end
end
end

```

```

---
- hosts: webservers
  gather_facts: false
  become: true

```

```
tasks:
  - name: Instalar apache
    ansible.builtin.apt:
      name: apache2
      state: present
```