

Programação para Internet (PPI)

Professor: Romulo Beninca

Aluna: Roberta Claudio

e-mail: romulo.beninca@ifsc.edu.br

- Lembre-se trabalhos com cópias parciais ou totais tem nota igual a zero.
- É recomendado a pesquisa em fontes externas principalmente documentações oficiais e livros, mas deve-se referenciar. Exemplo: De acordo com Pedro Zebo (2020), ou Bla bla bla bla bla (w3C, 2020).

Lembre-se de adicionar Referências caso utilize citação.

- **Para entrega da lista pelo moodle:** Arquivo ZIP contendo os exercícios separados em pastas.
- 1) Explique com um exemplo o que é o *model box* e as propriedades para manipulação de cada uma das camadas que fazem parte. Explique os fluxos existentes e o funcionamento, com exemplo de sua autoria.
- *Model box* possui 4 propriedades das quais se relacionam com as dimensões do elemento HTML. Os elementos são: **content**, **padding**, **border** e **margin**.

- **Content:** Área do conteúdo, onde é utilizada para colocar o conteúdo do html dentro, pode possuir cor de fundo, imagens, fonte.

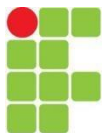
```
.exemploContent {  
background: #fff;  
width: 250px;  
height: 50px;  
font-style: normal;  
}
```

- **Padding:** É usada para controlar a área que irá preencher um elemento.

```
.exemploPadding {  
padding: 10px 20px 30px 40px;  
/* padding: 20px 0px 0px 20px;  
padding: 16px 10px */  
}
```

- **Border:** É usado para definir o estilo da borda de um elemento.

```
.exemploBorder {  
border: 2px #fff;  
border-radius: 5px; /* deixa o cantinho arredondado */  
border-style: solid;  
}
```



- **Margin:** É usada para controlar a margem de um elemento e influencia também no espaçamento entre os demais elementos ao redor.

```
.exemploMargin {  
margin: 10px 20px 30px 40px;  
/* margin: 20px 0px 0px 20px;  
margin: 16px 10px */  
}
```

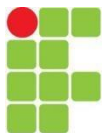
2) Com CSS é possível alterar o fluxo normal de uma página e definir o posicionamento de cada elemento, explique os fluxos existentes e o funcionamento, com exemplo de sua autoria.

- Pode-se alterar o fluxo normal de uma página e definir o posicionamento de cada elemento usando a propriedade **Position** e com suas respectivas palavras-chave, que são:
 - **Static:** Posicionado de acordo com o fluxo do documento, pode-se usar: top, right, left e bottom.

```
.Static {  
position: static;  
top: 10px;  
left: 40px;  
background: #fff;  
}
```

- **Relative:** Também é posicionado de acordo com o fluxo normal do documento, mas desloca-se em relação a si por causa dos valores acima, top, right, left e bottom.

```
.posicao {  
position: absolute;  
top: 10px;  
left: 40px;  
background: rgb(155, 28, 150);  
}
```



- **Absolute:** É removido do fluxo normal do documento, e se posiciona de acordo com o seu ancestral, caso tenha, se não ele é colocado em relação ao inicial containing block.

```
.posicao {  
  position: absolute;  
  background: yellow;  
  top: 30px;  
  left: 30px;  
}
```

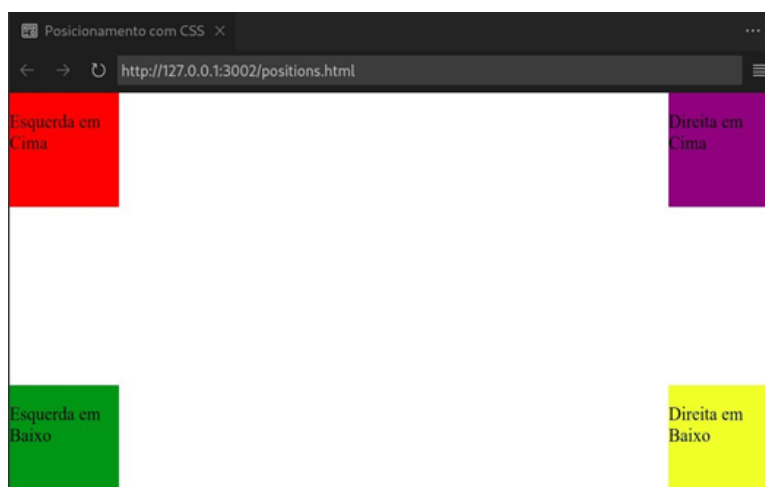
- **Fixed:** Também é removido do fluxo normal do documento e não cria nenhum espaço no layout da página. Também é colocado em relação ao containing block.

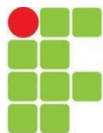
```
.fixex {  
  position: fixed;  
  top: 80px;  
  left: 10px;  
  background: blue;  
}
```

- **Sticky:** É posicionado de acordo com o fluxo normal do documento e desloca-se em relação ao ancestral de rolagem mais próximo e ao containing block.

```
.sticky {  
  position: sticky;  
  top: 10px;  
}
```

3) Utilizando a propriedade position, recrie um documento HTML e CSS com o seguinte visual





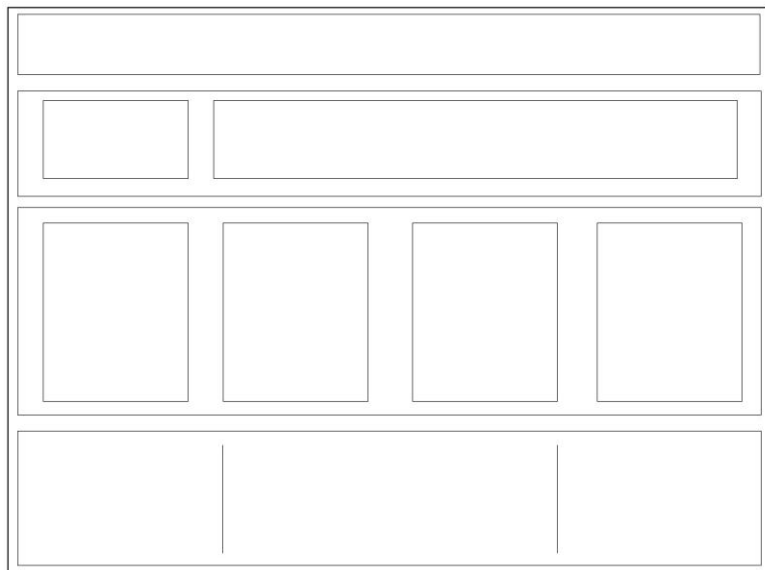
4) Utilizando a propriedade *position*, recrie um documento HTML e CSS com o seguinte visual

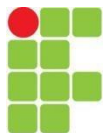


5) Recrie o seguinte layout utilizando Flex

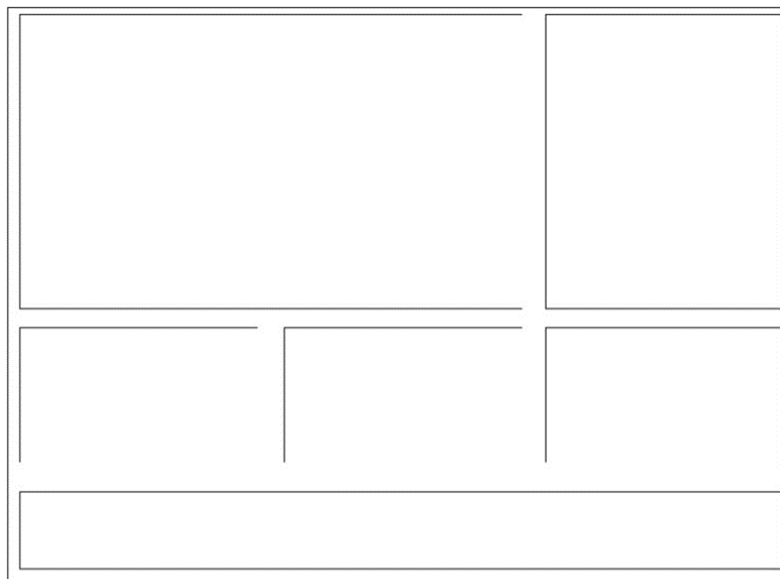


6) Recrie o seguinte layout utilizando Flex





7) Recrie o seguinte layout utilizando Flex



8) Crie dois exercícios práticos sobre o uso de Flexcontent e resolva seus exercícios. Use a criatividade.