



UNIVERSIDADE  
FEDERAL DO CEARÁ

# Linguagens de Marcação e Scripts

Prof. Jefferson de Carvalho

# Objetivos

---

- Objetivos Gerais

- Entender como usar scripts e linguagens de marcação para construir interfaces dinâmicas

- Objetivos específicos:

- Conhecer e aplicar conceitos de HTML e CSS
- Conhecer e aplicar conceitos de HTML5
- Conhecer e aplicar conceitos de Javascript
- Conhecer e aplicar conceitos de JSON

# Conteúdo Programático

- HTML

- Tags, atributos, elementos semânticos...

- CSS

- Seletores, cores, fundos, bordas...
- Layouts
- Componentes

- JavaScript

- HTML DOM
- JSON
- AJAX

- Bootstrap

```
60 //fires the appear event when appropriate
61 var check = function() {
62     //is the element hidden?
63     if (!t.is(':visible')) {
64         //it became hidden
65         t.appeared = false;
66         return;
67     }
68
69     //is the element inside the visible window?
70     var a = w.scrollLeft();
71     var b = w.scrollTop();
72     var o = t.offset();
73     var x = o.left;
74     var y = o.top;
75
76     var ax = settings.accX;
77     var ay = settings.accY;
78     var th = t.height();
79     var wh = w.height();
80     var tw = t.width();
81     var ww = w.width();
82
83     if (y + th + ay >= b &&
84         y <= b + wh + ay &&
85         x + tw + ax >= a &&
86         x <= a + ww + ax) {
87
88         //trigger the custom event
89         if (!t.appeared) t.trigger('appear', settings.data);
90
91     } else {
92
93         //it scrolled out of view
94         t.appeared = false;
95     }
96 };
97
98 //create a modified fn with some additional logic
99 var modifiedFn = function() {
100     //mark the element as visible
101     t.appeared = true;
102
103     //is this supposed to happen only once?
104     if (settings.one) {
105
106         //remove the check
107         w.unbind('scroll', check);
108         var i = $.inArray(check, $.fn.appear.checks);
109         if (i >= 0) $.fn.appear.checks.splice(i, 1);
110     }
111
112     //trigger the original fn
113     fn.apply(this, arguments);
114 };
115
116 //bind the modified fn to the element
117 $.fn.appear.one('appear', settings.data, modifiedFn);
118 settings.data, modifiedFn);
```

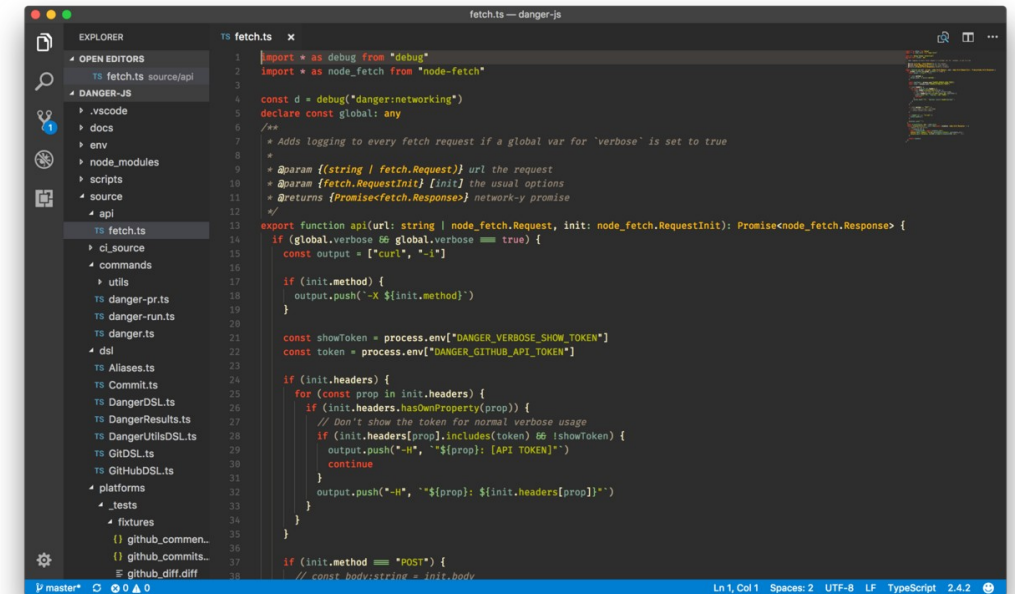
# Ferramentas

-  Editor de código: Vscode

- Avisos oficiais: SIGAA

-  Referência: w3schools.com

- Comunicação: e-mail ([jeffersoncarvalho@ufc.br](mailto:jeffersoncarvalho@ufc.br))



VSCode



# Avaliação

---

- Atividades + Provas
- Cada atividade tem seu peso. Se não estiver especificado, é peso 1.
- Entrega atrasada: 2 pontos de desconto por dia
- **Nota final =  $0,2 * (\text{média\_atividades}) + 0,8(\text{média\_provas})$**
- Se nota final maior ou igual a 7
  - Aprovado!
- Se nota final entre 4 e 7
  - Avaliação final
  - Aprovado se  $(\text{nota final} + \text{Avaliação})/2$  maior ou igual 5

# Dinâmica Atividades

---

- As atividades são feitas em sala e terminadas em casa
- Teremos atividade em sala na maioria das aulas
- Pode entregar atrasado com desconto de 2 pontos por dia
- Se perder alguma atividade, é melhor fazer a atividade perdida (mesmo que não valha nada) para treinar a matéria anterior
  - É difícil fazer uma atividade sem fazer as anteriores
  - O conhecimento em programação é uma escada
- Feedback das atividades é em sala
  - Individualmente, por demanda de vocês
  - Para todos, por correção da atividade em sala

# Observações importante!

---

- **Autonomia**
- Atividades são individuais
- Monitores e amigos podem ajudar pontualmente
  - Mas é necessário haver uma autonomia em fazer os trabalhos
  - Cuidado para não usar seus monitores como muletas
  - Trabalhos ficam super parecidos (quase cópia)
- É preciso **independência**! Afinal, a prova é individual
- **PS:** Cuidado com a propaganda negativa dos veteranos!

# Erros

---

- Erros fazem parte do aprendizado!
- É preciso saber resolver os erros sozinho.
- Melhor processo de aprendizado: entender o erro e tentar consertar. Se não conseguir, pede ajuda.
- Erro comum: Mudar o código na tentativa/erro sem entender o que está acontecendo



# Processo de resolução de erros

---

- O que fazer?
  - Ler mensagem de erro com calma
  - Traduzir para português
  - Tentar entender
  - Olhar o inspetor do navegador: verificar onde estão os elementos, se as regras css funcionaram...
  - `console.log` é seu melhor amigo! Imprimir variáveis na tela seguindo o fluxo da execução
- Se não conseguir
  - Olhar no google
  - Falar com o professor/monitor/amigo

---

# Perguntas?

Prof. Jefferson de Carvalho  
Prof. Victor Farias (slides cedidos)