

Administração Central Cetec Capacitações



Aula 09 - Biblioteca Core

1. Core.js

A biblioteca *Core* engloba todas as funções dentro de um objeto JavaScript, tornando-os métodos desse objeto, anteriormente as funções eram englobadas dentro de um objeto JavaScript usando um objeto literal.

```
var Core = {
    método01: funcion(..)
    {
        ...
    },
    método02: funcion(..)
    {
        ...
    },
}
```

Embora a *biblioteca Core* tenha a mesma estrutura do código, mostrado anteriormente, o código que utilizamos para definir o objeto e seus métodos usa:

```
var Core = {};
meuobjeto.método01:function(...)
{
...
};
meuobjeto.método02:function(...)
{
...
};
```



Administração Central Cetec Capacitações

O objeto criado neste exemplo é idêntico ao primeiro, o motivo de utilizarmos este estilo é permitir a liberdade de definir diferentes versões de cada método com base no navegador no qual o script está sendo executado.

2. Métodos receptores de evento

Existem dois modelos diferentes de implementar os receptores de eventos: Internet Explorer até e incluindo a versão 7 e o modelo DOM 2 do W3C suportado por todos os outros navegadores. A *biblioteca Core* preenche a lacuna, fornecendo um conjunto de métodos que funcionará nesses dois modelos:

- *Core.addEventListener()* designa um receptor da função/método como receptor dos eventos.
- *Core.removeEventListener()* remove um receptor da função/método anteriormente adicionado.
- *Core.preventDefault()* impede que o navegador execute uma ação padrão associada ao evento, um objeto evento passado como argumento para um receptor de eventos.
- Core.Propagation() impede que os receptores de eventos sejam ativados em resposta a eventos.

2. Partida do script

Usamos o método *start* da *biblioteca Core* para executar o método *init* de um script assim que o documento tenha terminado de carregar.

```
var script = {
    init: function(){
    },
};
Core.start(script);
```

3. Métodos de Gerenciamento de classe CSS

Os métodos a seguir da *biblioteca Core* estão relacionados a manipulação das classes CSS aplicadas aos elementos em seus documentos HTML.



Administração Central Cetec Capacitações

- Core.addClass(destino, aClasse) Adiciona na propriedade className de destino a classe
 CSS especificada (aClass), sem remover outras classes que já tenham sido utilizadas;
- *Core.getElementByClass(aClasse)* retorna um array de todos elementos documento que tenham classe CSS;
- *Core.hasClass(destino, aClasse)* retorna *true* se o destino tiver classe CSS especificada (*aClass*) aplicada e *false* se não tiver;
- Core.removeClass(destino, a Classe) Remove a propriedade className de destino a classe
 CSS especificada (aClass), sem remover quaisquer das outras classes que também possam ter sido aplicadas a destino.

Por último, o método para recuperar estilos calculados

Core.getComputed(elemento, propriedadeDoEstilo) – Recupera o valor da propriedade CSS Indicada por *propriedadeDoEstilo*, uma vez que todas as diversas fontes de estilos CSS tenham sido aplicadas a elemento.

4. Pequeno Exemplo.

Neste primeiro momento pode ser confuso!!!, mas fique atento nas cores e chamadas. Este exemplo abre a caixa de *alert*() quando o usuário clicar no link especificado em <a>.



Administração Central Cetec Capacitações

O Deu Bug?

- 1. Durante a capacitação, vamos utilizar outros exemplos para ficar o uso da biblioteca Core.js;
- 2. O link da biblioteca está disponível na plataforma;