

Introdução às Tecnologias Web

Guião da Aula Teórico-Prática sobre JavaScript (Parte 3)

Antes da aula

Nos tutoriais de JavaScript da W3Schools, ler o tutorial de [JavaScript Events](#) e o tutorial de JS HTML DOM, começando em [DOM Intro](#).

Durante a aula

Comece por obter a sua resolução dos exercícios da aula anterior, sobre o **jogo de adivinha o número**, assumindo-se que gera números aleatórios e pede números inteiros dentro do intervalo indicado na configuração, e que os botões são ativados e desativados corretamente.

O objetivo desta aula é acrescentar funcionalidades ao jogo, que requerem alterações no conteúdo HTML e, naturalmente, no código JavaScript. Podem, eventualmente, ser criadas novas regras CSS, mas tal não é obrigatório.

Exercício 1: Relembre os elementos e atributos em `index.html`, e as constantes, variáveis, e funções em `comportamento.js`. Analise que código é executado quando o *browser* termina o carregamento do documento HTML, bem como que funções são invocadas pelos botões da interface do jogo.

O jogo tem dois estados principais: a) de configuração de opções, que não permite jogar (o botão de fazer uma nova tentativa para adivinhar o número está desativado); e b) de jogo, que não permite configurar (os botões de mudar os valores das opções estão desativados, bem como o de gerar um número aleatório e começar um jogo). Atualmente, para um jogo terminar o utilizador tem de acertar no número ou então atingir o número máximo de tentativas.

Exercício 2: Em `index.html`, defina um botão para cancelar um jogo em curso, com nome `btnCancelaJogo` e inicialmente desativado, que invoca a função `cancelaJogo()` quando é premido. Em `comportamento.js`, crie duas constantes, `BOTAO_CANCELA_JOGO` e `RESULTADO_CANCELOU`, com valores apropriados (baseie-se nas constantes com nomes parecidos). Na nova função `cancelaJogo()`, chame `terminaJogo()` com argumento igual ao novo tipo de resultado. No *browser*, comece um jogo e tente cancelá-lo. Conseguiu? Se não, inspecione o código de `iniciaJogo()` e certifique-se que o botão de cancelar o jogo fica ativado aquando do começo de um jogo. Para manter a coerência, desative o botão na função `terminaJogo()`.

Para o próximo exercício, é útil conhecer a função `setInterval()`, que permite invocar uma outra função periodicamente, com o intervalo de tempo medido em milésimos de segundo, devolvendo um identificador de temporizador, que pode ser passado a `clearInterval()` para parar as invocações periódicas. Existe ainda `setTimeout()`, que serve para chamar uma só vez uma função depois de decorrido um período de tempo, também em milissegundos, e `clearTimeout()` para cancelar o temporizador.

Exercício 3: Acrescente um medidor de tempo de cada jogo, em segundos, entre a tabela de tentativas e o botão de fazer uma nova tentativa para adivinhar o número. Para tal, comece por criar um parágrafo HTML com o seguinte texto: o jogo decorre há 0 segundos, em que o valor 0 fica dentro de um elemento span com identificador `spanTempoJogo` (guarde este identificador numa nova constante `SPAN_TEMPO_JOGO`). Depois, adicione uma propriedade ao objeto na variável global `jogo`, chamada `inicio`, e inicialize-a com o valor da expressão `Math.floor(Date.now() / 1000)` em `iniciaJogo()`. A seguir, crie uma nova função `mostraTempoJogo()`, a qual deve ser invocada de segundo a segundo enquanto um jogo estiver a decorrer. O identificador do temporizador periódico deve ficar guardado na variável global `temporizadorTempoJogo`. Certifique-se que o medidor de tempo para quando o jogo chega ao fim.

Os próximos dois exercícios dizem respeito a uma mesma nova funcionalidade do jogo, nomeadamente de configurar a duração máxima permitida. O utilizador deve ser mantido a par do tempo que resta e, se este se esgotar, o jogo deve ser automaticamente cancelado. De notar que o limite de tentativas para adivinhar o número continua a ter de ser verificado.

Exercício 4: Inclua a opção de duração máxima do jogo, em segundos, na tabela de configuração. Para tal, comece por definir a constante `DURACAO_MAXIMA_OMISSAO` com o valor 60 (um minuto), bem como `DURACAO_MINIMA`, a valer 10. Depois, acrescente uma linha na tabela de configuração em que o elemento `td` que guarda o valor da opção tem identificador `tdDuracaoMaxima`, o botão da ação designa-se `btnPedeDuracaoMaxima`, e a função que invoca, se for premido, é `pedeDuracaoMaxima()`. Crie constantes para guardar esses identificadores, respetivamente `TD_DURACAO_MAXIMA` e `BOTAO_DURACAO_MAXIMA`. A seguir, adicione a propriedade `duracaoMaxima` ao objeto na variável global `configuracao`, inicializada com o valor por omissão anteriormente definido. Por fim, crie a constante `PROPOSITO_DURACAO_MAXIMA`, a qual deve ser usada no código da nova função `pedeDuracaoMaxima()`, nomeadamente na chamada a `pedeNumeroInteiro()`, e atualize o código de `iniciaJogo()` e `terminaJogo()`, por causa da ativação e desativação dos botões em cada estado do jogo, bem como de `mostraConfiguracaoJogo()`, para passar a ser mostrada a duração máxima do jogo na tabela de configuração.

Exercício 5: Mostre o tempo que falta até ao final do jogo, informando o utilizador a cada segundo que passa, e, se a duração máxima tiver sido atingida, cancele automaticamente o jogo. Baseie-se nos passos relevantes do exercício 3, com as seguintes adaptações: o parágrafo HTML passa a ser `tem mais 0 segundos para jogar`, em que o valor 0 fica dentro de um elemento span com identificador `spanTempoRestante` (guarde este identificador em `SPAN_TEMPO_RESTANTE`); a função a invocar periodicamente é `mostraTempoRestante()`; e o identificador do temporizador designa-se `temporizadorTempoRestante`. De notar que `mostraTempoRestante()` apenas é responsável por mostrar o tempo restante ao utilizador, devendo o jogo ser cancelado, por o tempo ter sido esgotado, por recurso a um outro temporizador, `temporizadorDuracaoMaxima`, que chama `terminaJogo()` uma só vez, com `setTimeout()`. Tenha o cuidado de, se o jogo terminar antes do tempo ter expirado, fazer `clearTimeout()`, caso contrário o temporizador continua ativo e vai cancelar um jogo posterior.