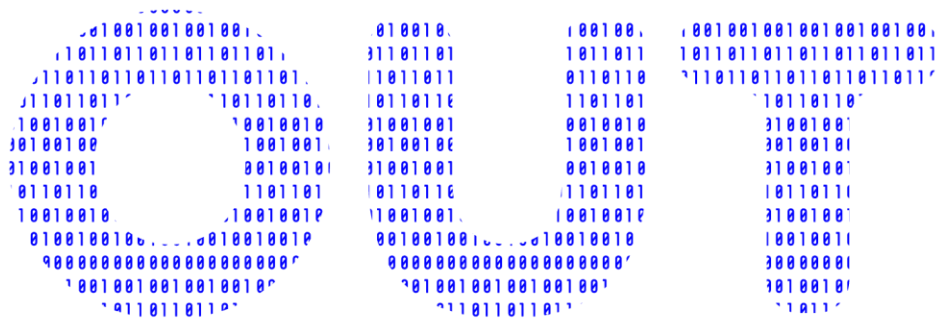




CODING -



> valuable outside tool
for employment

IO4. CODING-OUT - CURSO DE FORMAÇÃO PARA RECLUSOS

UNIDADE 7 DESENVOLVIMENTO DE APLICAÇÕES



Índice

Unidade 7. Subunidade 7.1	3
Pilares de programação web móvel.....	3
A Perspetiva dos Dispositivos	3
Unidade 7. Subunidade 7.2	17
Design&Usabilidade	17
Desenho Orientado à Usabilidade.....	17
Unidade 7. Subunidade 7.3	23
Autoaprendizagem.....	23
Exercícios.....	23
Exercícios Online.....	24
Unidade 7. Subunidade 7.4	25
Vamos praticar	25
Trabalho em equipa e atividade para trocar de papéis.....	25
Referências para o desenvolvimento de conteúdos:	26
Referências para o desenvolvimento da atividade:	27



Unidade 7. Subunidade 7.1

Pilares de programação web móvel

Tempo: 02h00m

A Perspetiva dos Dispositivos

Desktop/Laptop

A Web como a conhecíamos até recentemente tem como alvo principal os ecrãs maiores e processadores mais poderosos de desktops ou laptops, onde o utilizador geralmente está sentado e usando uma conexão rápida Ethernet ou de banda larga Wi-Fi. Este modo ainda é o padrão para pessoas que trabalham em escritórios, portanto, muitos sites Business-to-Business (B2B) são construídos para se adequar a essa configuração. Mas o uso do computador desktop está diminuindo, já que muitas pessoas hoje em dia, especialmente utilizadores domésticos, tendem a usar dispositivos móveis ou tablets (o que irei abordar em breve). A resolução do ecrã de 1024 × 768 está caindo e desuso, mas ainda é proeminente, enquanto as versões mais recentes do MacBook Pro da Apple apresentam uma resolução de 2880 × 1800. Os sites otimizados para o primeiro parecerão muito pequenos e ficarão um tanto perdidos com o segundo. O universo dos browsers para desktop inclui cinco opções principais: Chrome, Safari e Opera, que compartilham o motor WebKit (o Opera tinha seu próprio motor, Presto, que agora está sendo descontinuado); Firefox; e Internet Explorer. Felizmente, todos tendem a implementar recursos de uma maneira padrão e todos têm um ciclo de atualização frequente ou semi-frequente (ou estão se movendo em direção a ele), de modo que os utilizadores obtenham novos recursos com bastante rapidez.



Mobile

Ao mencionar dispositivo móvel, estamos geralmente a referenciar o telefone. A variedade é ampla: desde hardware mais antigo e mais barato executando browsers rudimentares até telefones com recursos mais sofisticados e dotados de browsers otimizados até smartphones potentes com browsers semelhantes aos que se encontramos nos laptops, bem como uma série de aplicativos que permitem a visualização de conteúdos. Obviamente, a vantagem destes dispositivos é apenas essa: sua mobilidade - a capacidade de encontrar coisas ao seu redor, obter instruções e pesquisar informações rapidamente quando estiver fora de casa ou do escritório. Porém, cada vez mais, os dispositivos móveis estão sendo usados em casa enquanto você assiste TV ou no trajeto matinal ou quando fica entediado em ficar na fila de trânsito. Na verdade, os dispositivos móveis têm menos a ver com estar em movimento e mais com estar sempre disponível. O espaço móvel está mudando mais rápido do que qualquer setor do mercado, já que muitos utilizadores trocam de dispositivos em ciclos de atualização de 12 a 18 meses e centenas (senão milhares) de novos modelos são introduzidos a cada ano.

Tablet

Os tablets já existem há vários anos, mas só após o lançamento do iPad, por parte da Apple, em 2010 é que se tornaram mais um gadget de lazer do que um acessório de trabalho. Muitos outros dispositivos seguiram o iPad, com tablets baseados em Android com dimensões de 7 "de tamanho médio e a Microsoft avançando com o Windows 8 fortemente focado em tablets. O tamanho generoso e a maior dependência de Wi-Fi (em muitos casos) torna os tablets portáteis em vez de realmente móveis; eles existem num contexto mais parecido com um laptop do que com um dispositivo móvel. A maioria das pessoas os usa em casa e, embora sejam carregados, geralmente são levados para fora e usados apenas quando o utilizador está parado, em vez de andar na rua. Como os smartphones, os tablets são táteis, dependendo da entrada por toque na maioria dos casos (embora alguns também aceitem entrada através de caneta). Pesquisas mostram que eles tendem a ser usados para entretenimento e navegação de conteúdos na Internet Os principais



browsers existentes nos tablets são essencialmente os mesmos que se encontram nos dispositivos móveis.

O Contexto

Um aspeto importante a se ter em mente é a inexistência de informação relevante sobre as pessoas que usam nossos sites. Não sabemos onde eles estão, não me refiro geograficamente, pois podemos usar o GPS para localizá-los. Quero dizer se eles estão em casa, no trabalho, no autocarro e assim por diante. Em essência, não temos ideia do contexto em que alguém está usando nosso site. A presunção comum é que as pessoas que usam o dispositivo móvel estão em movimento e com pressa, muitas vezes usando conexões de baixa largura de banda, mas nem sempre é o caso: o dispositivo móvel costuma ser usado em casa com uma boa conexão Wi-Fi rápida. Mas mesmo a conexão não conta toda a história; o sinal pode ser fraco ou a largura de banda reduzida devido ao congestionamento da rede. Além disso, presume-se que, para um dispositivo de ecrã grande como uma TV, os espectadores desejam um ambiente gráfico mais rico, mas este nem sempre é acompanhado de um processador potente ou de uma conexão rápida. Muitas TVs habilitadas para a web têm processadores não mais potentes do que um smartphone, e a velocidade de conexão está sujeita aos mesmos constrangimentos de qualquer outro dispositivo. E com o tamanho vem a questão da portabilidade. Quanto mais portátil o dispositivo, menos certeza podemos ter do contexto em que ele está sendo usado. Na verdade, não podemos fazer suposições.

“Rapidez” é o Único Contexto que Importa

Considere o seguinte desafio. Você não sabe quem são os seus utilizadores, onde estão, o que estão fazendo ou com qual dispositivo estão acedendo ao seu site. Você pode descobrir algumas dessas informações (embora não todas), mas seu contexto completo é completamente desconhecido e é altamente variável para cada indivíduo. A única coisa que você pode ter certeza razoável é que eles querem acesso ao que você está oferecendo ou querem descobrir que não é o que eles querem. De qualquer forma, eles querem uma resposta rápida. O desempenho é o único critério



que importa. Estejam os utilizadores em um smartphone durante a hora do ponta e procurando informações sobre o próximo comboio para casa ou navegando em um site de compras enquanto sentados confortavelmente no sofá em casa, eles têm uma tarefa que desejam concluir o mais rápido possível e concluí-la tarefa usando o dispositivo mais próximo fará com que eles se sintam mais eficientes (isso é conhecido como tempo encontrado).

Assim, o seu site precisa ser rápido - e rápido - independentemente do dispositivo em que está sendo exibido. E rápido significa não apenas desempenho técnico (que é extremamente importante), mas também a capacidade de resposta da interface e a facilidade com que os utilizadores podem navegar no site e encontrar o que precisam para concluir a tarefa que os trouxe até você.



Desenhar para Melhorar a Experiência de Todos os Utilizadores

Esta seção consiste na aprendizagem das técnicas fundamentais que permitem o desenho de sites otimizados independentemente do dispositivo do qual são acedidos. Uma capacidade importante a que os métodos têm procurado responder é a adaptabilidade e responsividade dos sites através de *media queries*.

A adoção generalizada de CSS permitiu que você deixasse para trás as limitações rígidas dos sites baseados em tabelas, e as responsividade e adaptabilidade vão um passo adiante, permitindo que você estilize as páginas de uma forma que seja compatível com as especificações de cada dispositivo em que seus sites são exibidos. Uma *media query* é uma declaração lógica: se a lógica for verdadeira, as regras de estilo dentro da declaração são aplicadas; se a lógica for falsa, as regras serão ignoradas. Os parâmetros das instruções são conhecidos como recursos de *media queries*, e os mais comumente usados atualmente dizem respeito às dimensões do dispositivo ou da janela de visualização.

```
<link rel="stylesheet" href="foo.css" media="screen">
```

Este código chama a folha de estilo externa foo.css apenas quando o dispositivo de visualização é um ecrã - em outras palavras, não um tipo de *media query* diferente, como impressão. Você estende essa sintaxe simplesmente adicionando a palavra e incluindo a própria consulta entre parênteses:

```
<link rel="stylesheet" href="foo.css" media="screen and (query)">
```

Esse código modificado tem duas condições: O tipo de *media query* deve ser um ecrã e a lógica da consulta de *media query* deve ser verdadeira. Se ambas as condições forem atendidas, foo.css será aplicado. Você também pode usar *media queries* para incluir folhas de estilo externas de outras folhas de estilo, usando a regra @import. O código a seguir tem a mesma lógica do bloco de código anterior, mas pode ser usado em tags de estilo ou em uma folha de estilo externa:

```
@import url('foo.css') screen and (query);
```



Por fim, você pode usar consultas de *media queries* in-line, o que é útil quando você deseja aplicar blocos de regras para casos específicos, em vez de chamar folhas de estilo externas:

```
@media screen and (query) { ... }
```

Recursos de *Media Queries* Baseados em Dimensões

As *media queries* são mais comumente usadas hoje para detetar as dimensões do dispositivo que está sendo usado para visualizar seu conteúdo e, em seguida, veicular as regras apropriadas para a sua exibição: texto grande e imagens para monitores grandes, texto pequeno e um layout de coluna única para smartphones, só para mencionar alguns exemplos. Claro, há mais *media queries* do que contextos simples, mas você entendeu a ideia geral. Você precisa considerar dois conjuntos de dimensões: primeiro, aquelas do próprio dispositivo e, segundo, aquelas da janela de visualização do agente (para a maioria das pessoas, é um browser, mas a janela de visualização também pode ser uma janela de uma aplicação) nesse dispositivo. Uma pessoa pode visitar seu site usando uma enorme televisão de ecrã amplo, mas isso não o preocupa se o aplicativo que a pessoa está usando para ver seu site ocupa apenas um quarto do ecrã. Em certos dispositivos, os dois conjuntos de dimensões são iguais - na maioria dos smartphones e tablets, por exemplo, a largura do browser é igual à largura do dispositivo.

As dimensões da janela de visualização são provavelmente as mais importantes e as que você mais usará, e os *media queries* que são relevantes para elas são altura e largura. Os recursos de dimensão da janela de visualização tomam como argumento um valor de comprimento único contra o qual a lógica é testada; se a dimensão em questão for igual ao valor do comprimento fornecido, a lógica é verdadeira e as regras são aplicadas. Na consulta a seguir, a regra dentro das chaves é aplicada ao elemento do corpo da janela de visualização quando ele tem exatamente 480px de largura:



```
@media screen and (width: 480px) {
  body { background-color: #00f; }
}
```

Usei um valor de px neste exemplo, mas qualquer unidade de comprimento é permitida. Independentemente de qual unidade você usa, um valor exato provavelmente é muito específico para a maioria das finalidades. Um par de ramais para o recurso o torna mais flexível, entretanto.

Muitos *media features*, incluindo largura, permitem os prefixos max- e min- antes do nome do recurso. Estes significam máximo e mínimo, respetivamente, que você provavelmente não precisa que eu explique, e na prática significam "não mais que" e "não menos que". Por exemplo, usando largura máxima, você pode aplicar regras de estilo a qualquer navegador com no máximo 480px de largura e usando largura mínima para qualquer navegador com largura não inferior a 480px:

```
@media screen and (max-width: 480px) { ... }
@media screen and (min-width: 480px) { ... }
```

Muitos *media features*, incluindo largura, permitem os prefixos max- e min- antes do nome do recurso. Estes significam máximo e mínimo, respetivamente, que você Se você tem certeza de que deseja trabalhar com as dimensões do dispositivo, não as da janela de visualização, você pode usar os recursos de largura e altura do dispositivo - embora, como antes, eu acho que largura é uma dimensão mais comumente usada do que altura, então vou me concentrar no primeiro. A maneira como você usa o recurso de largura do dispositivo é, para todos os fins práticos, a mesma que para o recurso de largura - apenas a métrica à qual ele responde é diferente. Tal como acontece com a largura, a largura do dispositivo pode ser estendida com prefixos máximos e mínimos:

```
@media screen and (max-device-width: 799px) { ... }
@media screen and (device-width: 800px) { ... }
@media screen and (min-device-width: 801px) { ... }
```

Outra forma de servir as regras depende da proporção do dispositivo ou da janela de visualização. A proporção da imagem é a proporção da largura para a altura (ou



largura do dispositivo para a altura do dispositivo); um monitor de 1024 × 768, por exemplo, tem uma proporção de aspeto de 4/3 - ou seja, quatro pixels horizontais para cada três pixels verticais - o que é comum para ecrãs mais antigos. Dispositivos *widescreen* mais recentes (como o iPhone) tendem a usar uma proporção de 16/9. Para segmentar um ecrã com proporção de 4/3, você deve usar o recurso de proporção de aspeto do dispositivo:

```
@media screen and (device-aspect-ratio: 4/3) { ... }
```

Como os prefixos max. e min também se aplicam a esses recursos, então, para criar uma consulta em que você deseja segmentar apenas janelas de visualização - não dispositivos - que estão atualmente em uma proporção de pelo menos 8/5, você deve usar a *feature* do aspeto mínimo -ratio:

```
@media screen and (min-aspect-ratio: 8/5) { ... }
```

Uma maneira rápida de descobrir quais relações de aspeto são maiores do que outras é dividir o primeiro algarismo pelo segundo; uma proporção de 4/3 resulta em 1,333, enquanto 8/5 resulta em 1,6. Quanto maior for o resultado, mais ampla será a proporção. O exemplo mostrado aqui se aplica apenas a ecrãs para os quais esse número é 1,6 ou mais. Provavelmente mais útil do que a figura muito precisa de uma proporção é a orientação da janela de visualização - seja no modo retrato (a altura é maior que a largura) ou no modo paisagem (a largura é maior que a altura). Você consulta isso com o recurso de orientação. Este recurso é especialmente útil para dispositivos portáteis como telefones e tablets que podem mudar de orientação com frequência:

```
@media screen and (orientation: portrait) { ... }
```

Os prefixos max- e min- não se aplicam a este recurso por razões óbvias.



Combinando e Negando *Media Queries*

Você pode adicionar uma série de instruções lógicas pelo uso repetido da palavra-chave e com *media queries* extras. Quando é usado, as regras são aplicadas apenas se toda a lógica das *media queries* for verdadeira; por exemplo, para testar se uma janela de visualização está no modo paisagem e pelo menos 800em de largura, você pode usar este código:

```
@media all and (orientation: landscape) and (min-width: 800em) { ... }
```

Você pode realmente simplificar mais; o tipo de todas as *media queries* é o estado padrão, portanto, você pode deixá-lo fora das consultas, tornando também a primeira instância de e desnecessária:

```
@media (orientation: landscape) and (min-width: 800em) { ... }
```

Você pode criar uma série de *media queries* e aplicar as regras se qualquer uma delas for verdadeira, usando o separador de vírgula para agir como um operador OU. Neste exemplo, as regras são aplicadas se o dispositivo estiver no modo paisagem ou se a largura da janela de visualização for de pelo menos 800em:

```
@media (orientation: landscape), (min-width: 800em) {}
```

Finalmente, você pode reverter a lógica de uma *media query* usando o operador NOT; adicionado ao início da consulta, este operador significa que as regras são aplicadas se a lógica da consulta for falsa. Neste código, as regras são aplicadas a qualquer dispositivo que tenha uma proporção diferente de 8/5:

```
@media not all and (device-aspect-ratio: 8/5) {}
```

Encontrar o equilíbrio certo de *media queries* para seu site pode ser um ato bastante complexo, envolvendo muitas variáveis diferentes, mas o resultado final pode ser extremamente satisfatório.



Uma Digressão Rápida: Tudo o que Precisa Saber sobre Pixels

À medida que avançamos para a era dos vários dispositivos, um ponto a ser considerado além das dimensões do ecrã é sua resolução. Muitos dispositivos novos, especialmente smartphones, agora são fornecidos com ecrãs de alta resolução, portanto, se você deseja fornecer uma experiência ideal para todos, deve considerar como as imagens e outros objetos serão exibidos nesses ecrãs. Antes de entrar nisso, no entanto, preciso falar rapidamente sobre pixels. Para falar sobre *media queries* dependentes de resolução, primeiro preciso definir alguns termos, começando com aqueles relacionados a pixels (presumo que os leitores saibam o que é um pixel). A contagem física de pixels de um ecrã é o número de pixels contidos na largura desse ecrã. Por exemplo, um ecrã de 1280 × 800 tem uma contagem física de pixels de 1280 ao longo de sua borda larga. O iPhone 3GS, em sua posição padrão (retrato), tem uma contagem física de 320 pixels, enquanto o iPhone 4 e superior, usando o ecrã “retina”, tem uma contagem de 640.

A densidade de pixels de um ecrã é tradicionalmente medida em pixels por polegada (PPI) - na verdade, se você preferir o sistema métrico, pode usar pixels por centímetro (PPC), mas vou me limitar à medida imperial para esta descrição. A medida imperial é às vezes conhecida como pontos por polegada (DPI), mesmo por software de edição de imagem de desktop popular, e os dois termos - PPI e DPI - podem ser considerados intercambiáveis. Independentemente disso, a densidade de pixels é uma medida do número de pixels que podem ser colocados em uma linha que se estende por uma polegada do ecrã do dispositivo.

Para a grande maioria dos ecrãs em uso até recentemente, o PPI é de 96. Com o aumento dos smartphones, no entanto, a contagem do PPI tem aumentado constantemente; o iPhone original (e todas as versões até o 3GS) tem uma resolução de 163 ppi, enquanto o iPhone 4 e superior tem o dobro disso, uns incríveis 326 ppi (apesar do ecrã permanecer do mesmo tamanho). A maioria dos ecrãs de alta densidade agora tem uma unidade de pixel virtual extra: o pixel independente de densidade (DIP), às vezes chamado de pixel CSS. Esta é uma unidade relativa; um pixel físico é igual a qualquer número de DIPs. Isso permite que você dimensione sites para cima (e para baixo), geralmente para fornecer melhor usabilidade em ecrãs menores.



A proporção de pixels virtuais para pixels físicos é conhecida como proporção de pixels do dispositivo. Vou usar a abreviatura DPR para isso, embora também seja conhecido como DPPX (pontos por pixel). Dispositivos que não têm DIPs têm um DPR de 1, ou seja, um pixel virtual para um pixel físico. Os dispositivos de alta resolução como o Samsung Galaxy S III ou o iPhone 5 têm um DPR de 2 - dois pixels virtuais para um físico. A figura abaixo ilustra isso. À esquerda está um bloco 2×2 de pixels físicos; no centro, um ecrã com DPR de 1,5 acomoda 9 DIPs no mesmo espaço; e à direita, um ecrã com um DPR de 2 encaixa 16 DIPs no espaço.



Media Queries para Resolução de Ecrã

Como você acabou de ver, muitos dispositivos agora vêm com ecrãs de altíssima resolução. Embora certamente seja uma coisa positiva, esses ecrãs têm uma ou duas desvantagens, principalmente em torno da exibição de gráficos. Por exemplo, JPG, PNG,..., são séries de pontos coloridos diferentes que são mais explicitamente vinculados a pixels que são salvos com uma configuração PPI. Portanto, um gráfico de *bitmap* que parece bom em um ecrã com um DPR baixo ficará com pouca qualidade em um dispositivo com um DPR alto. Isso significa que você provavelmente desejará usar bitmaps PPI mais altos para ecrãs de resolução mais alta, e é por isso que existem recursos para fazer exatamente isso.



O primeiro, e mais amplamente usado devido à popularidade do WebKit em dispositivos móveis, é um recurso proprietário chamado `-webkit-device-pixel-ratio`. Este recurso, junto com os prefixos máximos e mínimos relacionados, permite que você direcione o DPR de um dispositivo. Por exemplo, veja como adicionar uma imagem de alta resolução a dispositivos com um DPR de pelo menos 2:

```
E { background-image: url('foo.png'); }
@media screen and (-webkit-min-device-resolution: 2) {
  E { background-image: url('foo-hi-res.png'); }
}
```

Tudo muito simples, certo? Errado. `-webkit-device-pixel-ratio` é, como mencionado, uma extensão CSS proprietária do WebKit e não faz parte da especificação Media Queries. A especificação realmente contém uma consulta semelhante: resolução. Essa consulta melhora imediatamente a proporção de `-webkit-device-pixel-ratio` por ser mais flexível nos valores que aceita; você pode, por exemplo, usar a unidade de dpi. Lembre-se de que a resolução de um monitor padrão tem um DPI (ou PPI) de 96; sendo esse o caso, você pode direcionar esses monitores com este código:

```
@media screen and (resolution: 96dpi) {
  E { background-image: url('foo.png'); }
}
```

Mas você também pode atingir ecrãs de resolução mais alta multiplicando a contagem padrão, 96, pelo DPR. Por exemplo, a consulta a seguir se aplica a dispositivos com um DPR de 2 ou superior, assim como no exemplo `-webkit-device-pixel-ratio` alguns parágrafos atrás:

```
@media screen and (min-resolution: 192dpi) {
  E { background-image: url('foo-hires.png'); }
}
```

Você pode fazer isso de uma maneira ainda melhor, no entanto, usando a unidade `dppx`. Esta unidade é equivalente ao DPR, portanto, uma versão atualizada do exemplo anterior emprega este código:



```
@media screen and (min-resolution: 2dppx) {
  E { background-image: url('foo-hires.png'); }
}
```

-webkit-device-pixel-ratio existe desde o lançamento do iPhone em 2006, e uma grande quantidade de dispositivos legados estão por aí. Assim, caso pretenda que o seu site seja inclusivo para dispositivos mais recentes, você provavelmente terá que estender as *media queries* com um pouco de lógica extra. Conforme mencionado anteriormente, o uso de uma lista separada por vírgulas de *media queries* significa que as regras serão aplicadas se qualquer uma delas for verdadeira:

```
@media screen and (min-resolution: 2dppx),
  screen and (-webkit-min-device-resolution: 2) {
  E { background-image: url('foo-hires.png'); }
}
```

Essa lógica testa se o dispositivo tem um DPR de pelo menos 2, mas funciona em browsers WebKit legados e em aqueles que estão em conformidade com as especificações.

Adaptação ao Dispositivo

Os browsers móveis são capazes de dimensionar sites para melhor apresentação em ecrãs pequenos, mas presumem que a maioria dos sites é voltada para browsers de desktop. Essa suposição significa que, por padrão, eles mostram um site com zoom reduzido/reduzido para acomodar tudo (ou o máximo possível) no ecrã. Isso geralmente é conhecido como janela de visualização de layout, e a desvantagem da janela de visualização de layout é que a visualização reduzida pode deixar os sites otimizados para dispositivos móveis com uma aparência realmente muito pequena.

A solução é definir os parâmetros da janela de visualização em CSS. Você faz isso com a regra @viewport. Onde a meta tag viewport usa parâmetros diferentes no atributo content, @viewport usa sintaxe CSS para executar as mesmas funções com um subconjunto de propriedades especializadas. Para ilustrar, repeti o exemplo anterior:



```
@viewport { width: 480px; }
```

A razão pela qual esta solução é mais útil é que você pode combinar `@viewport` com *media queries* para ter diferentes parâmetros de janela de visualização, dependendo do dispositivo de visualização. Considere este exemplo:

```
@media screen and (max-width: 480px) {  
  @viewport {  
    width: 480px;  
  }  
}  
@media screen and (min-width: 481px) {  
  @viewport {  
    width: device-width;  
  }  
}
```

Para ecrãs com largura de 480px ou inferior, a janela de visualização será exibida em 480px e, em ecrãs maiores, em seu tamanho natural. Essa abordagem é conhecida como adaptação de dispositivo e tem a vantagem de tornar as alterações em todo o site muito mais fáceis, exigindo a modificação de apenas uma única regra CSS, em vez de um número desconhecido de páginas individuais.



Unidade 7. Subunidade 7.2

Design&Usabilidade

Tempo: 02h00m

Desenho Orientado à Usabilidade

Embora haja uma certa liberdade artística ao criar um site, também existem “melhores práticas” bem estabelecidas entre os web designers profissionais. Você já visitou uma página da web que era difícil de navegar, difícil de entender ou simplesmente pouco apelativa? Um bom web designer pode olhar essas páginas e oferecer sugestões para melhorias.

Nesta seção, você aprenderá algumas técnicas para tornar seu site o mais utilizável possível. Ao tornar o seu site fácil e divertido para os visitantes navegarem, você aumenta o tempo que as pessoas passam no seu site e o número de vezes que elas retornam.

O CSS também fornece métodos para controlar como os documentos serão apresentados em contextos diferentes do browser de desktop tradicional, como na impressão e/ou em dispositivos com ecrã reduzido. O CSS também possui regras para especificar a apresentação não visual de documentos, por exemplo, como eles serão apresentados via áudio. As folhas de estilo também são uma ótima ferramenta para automatizar a produção, pois você pode alterar a aparência de um elemento em todas as páginas do seu site editando um único documento de folha de estilo. As folhas de estilo são suportadas por todos os browsers modernos.

Compreendendo a Usabilidade

Usabilidade refere-se à experiência que os visitantes têm ao visualizar seu site. Inclui essas qualidades, resumidas da seguinte forma:

- *Facilidade de aprendizagem*, com que rapidez as pessoas entendem como funciona a navegação no site? As pessoas que nunca viram a interface podem



aprendê-la bem o suficiente para se orientar sem uma curva de aprendizado acentuada?

- *Facilidade de uso*, depois que as pessoas descobrem como navegar no site, é fácil para elas realmente encontrar as informações de que precisam? Um site altamente utilizável coloca as informações ao alcance dos visitantes, com ferramentas de pesquisa e navegação flexíveis e poderosas.
- *Capacidade de memorização*, quanto um visitante se lembrará recorrentemente do seu site? Um site altamente utilizável fica gravado na mente dos visitantes.
- *Tratamento de erros*, com que frequência os visitantes cometem erros ao navegar em seu site e quão fácil é para eles voltarem ao caminho certo? Um site altamente utilizável fornece mensagens de erro úteis quando ocorrem problemas, completas com hiperlinks que ajudam os utilizadores a fazer o que pretendem.
- *Satisfação subjetiva*, quanto as pessoas gostam de visitar seu site? Um site altamente utilizável é divertido de explorar.

A usabilidade é extremamente importante para gerar visitantes fiéis e recorrentes ao seu site. Uma campanha publicitária pode atrair visitantes ao seu site inicialmente, mas se o site não for fácil de usar, a maioria deles nunca retornará.

Planear para a Usabilidade

O planejamento da usabilidade deve começar antes de você criar a primeira página do seu site, porque suas respostas às principais perguntas sobre o propósito e o público-alvo irão moldar a organização geral e o layout do site. Em primeiro lugar, por que você quer um site? O que você espera alcançar com isso? Um site projetado para vender produtos on-line provavelmente será muito diferente de outro que visa principalmente o compartilhamento de informações. Esclarecer seus objetivos antes de começar a projetar seu site torna mais fácil definir e criar o site que você deseja.

Tente tornar suas metas específicas e vinculá-las aos objetivos de seu negócio ou organização. (por exemplo, as metas podem ter sido, por exemplo, reduzir chamadas



de clientes que precisam de ajuda, para fazer os visitantes se sentirem mais confiantes sobre a compra de produtos e / ou serviços, ...). Em seguida, qual público você está almejando? “Todos” é uma resposta pobre a essa pergunta. Você não pode agradar a todos e, se tentar, não agradará a ninguém. Pense nas características das pessoas que você está alvejando e no que elas procuram em um site.

Estruturação do Site

A próxima etapa é esboçar um gráfico mostrando como os utilizadores acedem ao conteúdo, começando com sua página inicial no topo. Todas as páginas que serão diretamente acessíveis na página inicial serão exibidas no primeiro nível, e as páginas subordinadas a elas serão exibidas nos níveis inferiores.

Aqui estão algumas dicas para planejar a organização do site:

- Decida os links que estarão na barra de navegação. Organize os links em ordem de importância de cima para baixo (ou da esquerda para a direita). A página inicial deve ser sempre o link mais alto ou mais à esquerda na barra de navegação.
- Decida qual conteúdo você entregará em cada página. Elimine todas as páginas que os visitantes não desejam ou que não fornecem informações que apoiem a meta de sua empresa ou organização para o site.
- Planeie maneiras de reduzir o número de cliques que o utilizador precisa de fazer em média. Uma maneira de fazer isso é colocar um link direto na página inicial para o conteúdo mais popular.

Planear um Layout Consistente

Para facilitar a navegação, todo o site deve ter um layout consistente, com elementos comuns como a barra de navegação e o título da página no mesmo local em cada página. A maneira mais simples de fazer isso é criar uma página de modelo e, em seguida, basear todas as outras páginas nela. Sua página de modelo pode usar tabelas, quadros ou divisões para o layout.



Seguem-se algumas dicas para o layout de sua página de modelo:

- Coloque um cabeçalho na parte superior da página, contendo o logotipo e o nome da sua organização.
- Posicione a barra de navegação à direita, à esquerda ou parte superior da página. As barras esquerda e superior são as mais comuns, mas muitos especialistas em usabilidade dizem que uma barra de navegação à direita é, na verdade, mais intuitiva para o visitante usar.
- Se você tiver um site muito rico em informações, considere ter várias barras de navegação - uma no topo da página para incluir as categorias principais gerais e outra à esquerda ou direita com uma lista mais longa de subcategorias.
- Faça a barra de navegação se destacar de alguma forma. Pode ser de uma cor diferente, ter um fundo diferente ou estar rodeado por uma caixa, por exemplo.
- Coloque uma barra de navegação somente texto na parte inferior da página para que as pessoas não precisem efetuar *scroll* de volta para o topo para navegar para outras páginas.
- Se você decidir usar molduras, tenha muito cuidado. É muito fácil criar um conjunto de quadros no qual um quadro é muito pequeno e corta o conteúdo inserido nele. Certifique-se de que cada *frame* tenha o tamanho adequado não apenas para o conteúdo padrão com o qual começa, mas para cada página que pode aparecer em cada *frame* durante toda a visita do utilizador ao seu site.
- Quando possível, torne o tamanho da página flexível (por exemplo, deixando uma coluna da tabela ou uma divisão vertical para preencher o espaço restante na janela). Se você estiver especificando uma largura fixa para o conteúdo da página, não faça com que ela tenha mais de 800 pixels de largura. Dessa forma, mesmo pessoas com ecrãs de baixa resolução poderão visualizá-lo sem necessidade de *scroll*.



- Selecione cores que reflitam o conteúdo e a identidade do site. Vermelhos e amarelos criam entusiasmo; azuis e verdes são calmantes. O corpo do texto deve ser composto por letras escuras em um fundo de cor clara.
- Adapte as opções de cores ao seu público-alvo. A pesquisadora Natalia Khouw relata, por exemplo, que os homens preferem o azul e o laranja, enquanto as mulheres preferem o amarelo e o vermelho. Os jovens gostam de cores primárias brilhantes; pessoas de meia-idade e mais velhas gostam de cores suaves, como prata, azul-acinzentado e amarelo claro.
- Selecione uma letra simples e legível como padrão, como Arial (Helvetica, Sans Serif), em um tamanho adequado para o seu público.

Planear o Contexto das Páginas Individuais

Depois de criar o modelo que formará a estrutura de cada página, comece a pensar no conteúdo exclusivo de cada página. Aqui estão algumas dicas para criar páginas da web eficazes:

- Use frases curtas (20 palavras ou menos) e parágrafos curtos (5 frases ou menos).
- Certifique-se de que haja algum espaço vertical entre cada parágrafo. Por padrão, a *tag* <p> deixa uma boa quantidade de espaço, mas algumas pessoas removem ou diminuem o espaço vertical modificando o estilo.
- Sempre que possível, divida as informações em listas com marcadores ou numeradas para facilitar a leitura.
- Combine o comprimento da página com seu propósito. As páginas que resumem ou fornecem navegação devem ser curtas; páginas que fornecem informações detalhadas sobre um assunto podem ser tão longas quanto necessário.
- Mantenha os artigos em uma página. Não divida o texto de um artigo em várias páginas apenas porque uma página parece longa. Os visitantes que desejam imprimir o artigo acharão muito mais fácil fazê-lo se estiver tudo em uma página e apreciarão não ter que clicar em um link para ver o resto do artigo.



- Divida artigos longos usando muitos títulos descritivos. Se o artigo tiver mais do que algumas páginas, inclua hiperlinks de favoritos na parte superior da página que apontam para os cabeçalhos principais.
- Se o conteúdo atinge mais de um nível de profundidade em seu site, use *breadcrumbs* para ajudar a navegação dos utilizadores no site. *Breadcrumbs* são uma sequência de hiperlinks que permitem ao utilizador ter a noção do nível onde se encontra na estrutura do site. Exemplo: Home> Jazz> John Coltrane
- Limite o tamanho dos arquivos gráficos que você usa em uma página para que o download da página não demore muito em uma conexão lenta. O tamanho total do arquivo de todos os gráficos em uma página não deve, idealmente, exceder 30 KB. Se você precisar mostrar gráficos maiores e de alta resolução, considere o uso de miniaturas.
- Procure maneiras de reduzir o espaço não utilizado. Em uma página com muito espaço vazio à direita, por exemplo, considere adicionar uma caixa de texto contendo informações. Uma maneira de fazer isso é usar uma divisão com posicionamento absoluto.



Unidade 7. Subunidade 7.3

Autoaprendizagem

Tempo: 07h00m

Exercícios

Inicie o processo visualizando a seguinte [página web](#) e o [CSS](#) correspondente em uma janela pequena para simular como seria em um dispositivo móvel. Ajuste um pouco do CSS para ajustar como a página é exibida.

- Aumente o tamanho da letra
- Diminua o tamanho da letra em h1
- Aumente o espaçamento entre linhas
- Adicione uma pequena margem às bordas da página

Usando o Google Chrome, por exemplo, você pode descobrir facilmente o tamanho da janela do seu browser usando o Web Inspector. Abra o inspetor da web no Chrome e redimensione a janela do navegador enquanto observa o canto superior direito. As dimensões em pixels devem aparecer conforme você redimensiona.

Comece a expandir a largura da janela do seu browser. Faça isso até chegar a um ponto em que o design não funcione mais. Por exemplo, o comprimento da linha pode se tornar muito longo para que o texto seja facilmente legível. Ou talvez a página se torne larga o suficiente para que não faça mais sentido que as imagens sejam exibidas em uma pilha.

Registe a largura da janela do browser neste ponto. Este será seu primeiro ponto de interrupção. Um ponto de interrupção é simplesmente um ponto em que definimos que o design deve ser alterado.

Crie uma nova *media query* no seu browser cujo valor é o correspondente à dimensão de pixel que você registou no teste anterior. Atribua-o como sendo a



largura mínima. Dentro desse ponto de interrupção, altere seu CSS para fazer o seguinte:

- Altere a cor de fundo (para ajudá-lo a ver quando a *media query* entra em vigor)
- Ajuste o tamanho da letra, se necessário
- Ajuste as margens da página
- Faça as fotos serem exibidas em duas colunas

Redimensione a janela do browser novamente para encontrar outro ponto de interrupção. Adicione uma nova *media query* em seu CSS e ajuste as seguintes regras:

- Mude a cor de fundo
- Ajuste o tamanho da letra, se necessário
- Ajuste as margens da página
- Faça com que as fotos sejam exibidas em três colunas
- Adicione uma largura máxima à página para garantir que o comprimento da linha nunca fique muito longo

(Sugestão de resolução: [HTML](#) e, [CSS](#))

Replique o processo acima mencionado na sua página pessoal.

Exercícios Online

https://www.w3schools.com/css/css_rwd_intro.asp



Unidade 7. Subunidade 7.4

Vamos praticar

Tempo: 04h00m

Trabalho em equipa e atividade para trocar de papeis

Os formandos são divididos em equipas de 2 ou 3 membros.

Cada membro da equipa deve validar o HTML e o CSS implementados pelo(s) colega(s) de equipa no exercício anterior, fornecendo ideias e/ou sugestões para melhoria, e de seguida, cada membro da equipa deve atualizar as suas páginas em conformidade.

Finalmente, cada formando pode apresentar as suas próprias páginas e discuti-las com os outros colegas.



Referências para o desenvolvimento de conteúdos:

Robbins, Jennifer (2012), *“Learning web design: a beginner's guide to HTML, CSS, Javascript, and web graphics”*. Sebastopol, CA: O'Reilly.

Wempen, Faithe. (2011), *“HTML5 : step by step.”*, Sebastopol, CA: Microsoft O'Reilly distributor.

Gasston, Peter (2013), *“The modern Web: multi-device Web development with HTML5, CSS3, and JavaScript”*. San Francisco, CA: No Starch Press.



Referências para o desenvolvimento da atividade:

Robbins, Jennifer (2012), *“Learning web design: a beginner's guide to HTML, CSS, Javascript, and web graphics”* . Sebastopol, CA: O'Reilly.

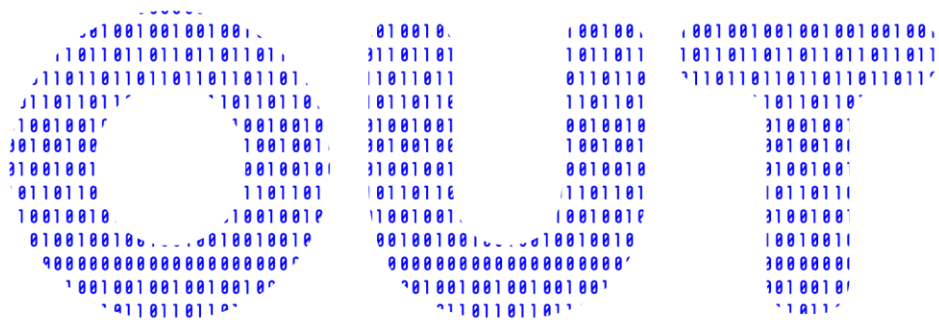
Wempen, Faithe. (2011), *“HTML5: step by step.”*, Sebastopol, CA: Microsoft O'Reilly distributor.

Gasston, Peter (2013), *“The modern Web: multi-device Web development with HTML5, CSS3, and JavaScript”*. San Francisco, CA: No Starch Press.

<http://web.simmons.edu/~grovesd/comm328/>



CODING-



> valuable outside tool for employment

IO4. CODING-OUT - CURSO DE FORMAÇÃO PARA RECLUSOS

UNIDADE 7 DESENVOLVIMENTO DE APLICAÇÕES