

情報基礎 Webページ作成

1

課題: Webページ作成

- ◆ 必要条件
 - ◆ index.htmlを含めてHTMLファイル3つ以上作成
 - ◆ 他人の作成したページに1つ以上リンクをはる
 - ◆ CSSファイル1つ作成
 - ◆ 画像を1つ以上表示
 - ◆ HTML/CSSチェックに合格
- ◆ ディレクトリの整理(オプション)
- ◆ 学内・学外公開(オプション)

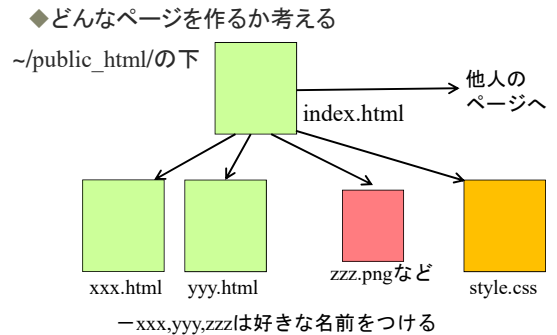
2

課題: Webページ作成

- ◆ この資料は3回分の資料です
- ◆ 提出(採点)は3回目の1週間後予定です
- ◆ 大体の目安として1日目、2日目、3日目と書いてありますが、できる人は気にせず進んで結構です
- ◆ 今は1日目です

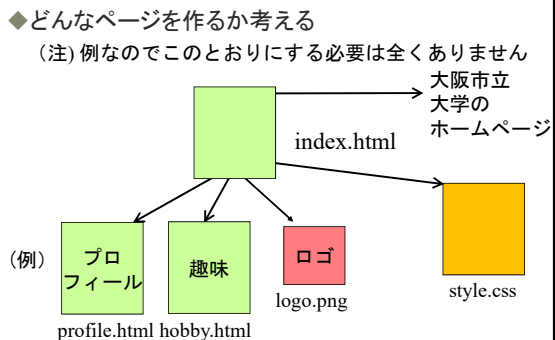
3

課題



4

課題



5

課題の手順(準備)

- ◆ エディタとブラウザを利用
- ◆ エディタの文字コード
- ◆ メモ帳の場合
 - ◆ 文字コード: UTF-8
- ◆ TeraPadの場合
 - ◆ 文字コード: UTF-8N
 - ◆ 改行コード: CF+LF

6

課題の手順

- ◆ 1. ディレクトリ(フォルダ)「public_html」を作成
- ◆ 以下はすべてpublic_html下で行う
- ◆ 「_」はアンダーバー(下線)です。通常はShiftキーを押しながら、右のシフトキーの左隣のキーを押します

7

課題の手順

- ◆ 2. index.html(トップページ)の作成
 - ◆ (1) 雛形(hinagata.html)ファイルを「名前をつけて保存」(コピーペーストでもよい)
 - ◆ HTMLのみ、ファイル名はindex.html
 - ◆ (2) エディタで<title></title>の間にタイトルを書いて保存、ブラウザで表示(更新)
 - ◆ (3) エディタで<body></body>の間に本文を書いて保存、ブラウザで表示(更新)
 - ◆ 以下は基本的には(3)の繰り返し

8

HTML 最小限必要な記述

```
<html>
<head>
  <title>(ここにタイトルを書く)</title>
</head>
<body>
  (ここに本文を書く)
</body>
</html>
```

HTMLファイル

9

HTMLの雛形

```
<!DOCTYPE html>          (意味)HTML宣言
<html lang="ja">          (意味)言語は日本語
  <head>
    <charset="UTF-8">      (意味)文字コードはUTF-8
    <title>(ここにタイトルを書く)</title>
  </head>
  <body>
    (ここに本文を書く)
  </body>
</html>
```

10

HTML

- ◆ HTML(Hyper Text Markup Language)とは
 - ◆ WWW記述言語
 - ◆ 文章の論理構造を要素(タグ)を使って記述
 - ◆ 2種類
 - ◆ <tag> ... </tag>
 - ◆ <tag>
- ◆ 特徴:
 - ◆ テキストファイルだけで記述できる
 - ◆ 細かいレイアウトにこだわらなくてよい
 - ◆ レイアウトを指定できるが強制はできない

11

HTML

- ◆ 注意
 - ◆ 空白、TABコード、改行コードは表示に影響なし
 - ◆ 大文字と小文字を区別しなくても表示は変わらない

12

HTML よく使われる要素

- ◆見出し
 - ◆<hx> ... </hx> (x=1,2,3,4,5,6)
- ◆段落
 - ◆<p> ... </p>
- ◆罫線
 - ◆<hr>
- ◆改行
 - ◆

13

HTML よく使われる要素

- ◆コメント(2回目以降でよい)
 - ◆<!-- ... -->
- ◆見たまま表示(使わなくてもよいです)
 - ◆<pre> ... </pre>

14

HTML よく使われる要素

- ◆リンク(アンカー)
 - ◆趣味
 - ◆地図
 - ◆大阪市立大学
 - ◆ログイン名の人のホームページ(後半でOK)

15

HTML よく使われる要素

- ◆画像の表示(2回目以降でよい)
 - ◆
 - ◆

16

HTML よく使われる要素

- ◆箇条書き(2回目以降でよい)
 - ◆
 - りんご
 - みかん
 - ◆
 - りんご
 - みかん

17

HTML よく使われる要素

- ◆表(簡単な例)(2回目以降でよい)
 - ◆<table>
 - ◆<tr><td>...</td><td>...</td></tr> 2列の場合
 - ◆...
 - ◆</table>

18

課題の手順

- ◆ 3. CSSファイルの作成
 - ◆ style.cssというファイルを作成
 - ◆ index.htmlからリンクをはる
 - ◆ <head></head>の中; <title>の下あたり
 - ◆ `<link rel="stylesheet" href="style.css">`

19

課題の手順

- ◆ 4. HTMLとCSSの動作の確認
 - ◆ 背景色やフォントの色を変えてみる
 - ◆ 背景色の場合 (CSS)
 - ◆ (例) body
 - ◆ `background-color: red;`

20

CSSとは

- ◆ Cascading Style Sheetの省略表記。シーエスエスと読む
- ◆ Webページのレイアウト(視覚的構造)を定義する「スタイルシート」の規格の一つ
- ◆ Webの標準化団体であるW3C(World Wide Web Consortium)によって標準化

21

W3Cで推奨される考え方

- ◆ 論理構造: マークアップ言語
 - ◆ HTML, XHTML, XMLなど
- ◆ レイアウト(見た目): スタイルシート言語
 - ◆ CSSなど

22

CSSのメリットの例

- ◆ 「New」という文字列の色を赤から黄色にかえたい場合
- ◆ 100ページ(100個のHTMLファイル)のWebサイト

23

HTML+CSSで記述

◆ HTML

```
<span class="attention">New</span>
```

“attention (名前は自由)” というclassを設定
100個のHTMLファイルは変更不要

◆ CSS (別ファイル)

```
◆ .attention  
◆ color: yellow;
```

1個のCSSファイルにおいて “attention”の色を
redからyellowに変更

24

CSSの基本

- ◆セレクタ {プロパティ:プロパティ値;}
- ◆(例)
- ◆p {color: #ff0000; }
- ◆<p>要素の色を赤に設定

25

CSS よく使われる例

- ◆背景色
 - ◆ background-color: 色;
- ◆背景画像(使わなくてもよい)
 - ◆ background-image: url(ファイルへのパス);
- ◆フォントの色やサイズ
 - ◆ color: 色;
 - ◆ font-size: サイズ;

26

CSS よく使われる例

- ◆中央揃え(ボックス)(2回目以降でよい)
 - ◆ margin: 上下幅 auto;
- ◆中央揃え(文字列)
 - ◆ text-align: center;
- ◆左揃え(文字列)(2回目以降でよい)
 - ◆ text-align: left;

27

CSS よく使われる例

- ◆幅(ボックス)(2回目以降でよい)
 - ◆ width: 幅;
- ◆下線(使わなくてもよい)
 - ◆ border-bottom: 種類 太さ 色;
- ◆上の罫線(使わなくてもよい)
 - ◆ border-top: 種類 太さ 色;

28

見本

- ◆HTML5+CSS2
 - ◆<https://www.ecs.osaka-cu.ac.jp/~fee26852/>
 - ◆index.html, profile.html, hobby.html
 - ◆css/style.cssほか
- ◆タグの使い方説明用のため、簡単にしています。一部いい加減です
- ◆授業の進行にあわせて内容が変わります

29

諸注意

- ◆ファイル名のつけ方
 - ◆半角英数字 例) index.html, link.html
 - ◆日本語や特殊記号(#とか)や空白を含まない
 - ◆種類に応じた拡張子を付ける 例) index.html, memo.txt, photo.gif
- ◆HTML/CSSファイル
 - ◆プレーンテキストとして作成
- ◆ファイルの場所
 - ◆public_html/の下に置く

30

諸注意

- ◆ 大文字と小文字
 - ◆ タグについては大文字でも小文字でもOK (例<html>と<HTML>)
 - ◆ 他のファイルにリンクを貼るときは大文字と小文字を間違えない 例) picture.gif に対してとしない
- ◆ イメージデータ形式
 - ◆ PNG, JPEG, GIF形式など
- ◆ 大きなデータ
 - ◆ 大きすぎるデータは極力おかない
 - ◆ おく場合は、データサイズを書いておくと親切

31

2日目

32

課題の手順

- ◆ 5. 新しいHTMLファイルの作成・編集
 - ◆ index.htmlを別名で保存して編集すると楽
 - ◆ (例) エディタで内容を編集してprofile.htmlという名前で保存
- ◆ 6. HTMLファイル同士でリンクをはる
 - ◆ (例) index.htmlからprofile.htmlへのリンク
プロフィール
 - ◆ (例) profile.htmlからindex.htmlへのリンク
トップ

33

課題の手順

- ◆ 7. 画像ファイルの作成
 - ◆ Paint, Paint 3D, GIMP等を用いてイラストやロゴを作成、GIF/JPG/PNG形式で保存
- ◆ 8. 画像ファイルの表示
 - ◆ HTMLでは (画像の表示)
 - ◆ CSSでは background-image: URL (背景画像の表示)

34

ブロックレベル要素とインライン要素

- ◆ ブロックレベル要素
 - ◆ 文書や段落を構成する基本要素
 - ◆ 例) 見出し要素、div要素
- ◆ インライン要素
 - ◆ 特定の部分に何らかの役割を持たせる要素
 - ◆ 中にブロック要素を含むことができない
 - ◆ 例) アンカー要素、span要素

35

idとclass

- ◆ id
 - ◆ 要素に固有の(一つしかない)identifier(名前)をつける
 - ◆ 1ページに1回しか出現できない
- ◆ class
 - ◆ 要素にクラス(分類)をつける
 - ◆ 1ページに2回以上出現できる

36

HTML+CSS よく使われる要素

- ◆ article
- ◆ section
- ◆ main
- ◆ header
- ◆ footer
- ◆ nav
- ◆ など
- ◆ 使い分けがよくわからなければ無理に使わなくてもよいです

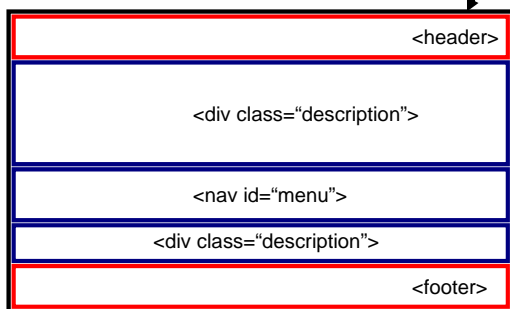
37

見本(山田太郎)

- ◆ HTML5+CSS2
 - ◆ <https://www.ecs.osaka-cu.ac.jp/~fee26852/mihon/>以下
 - ◆ index.html, profile.html
 - ◆ css/style.css
 - ◆ image/photo.gif
- ◆ コピーして使用してもよい
- ◆ 注)HTML5のIE対策はしていない

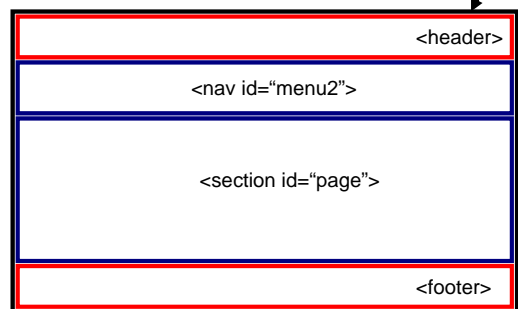
38

見本(山田太郎)のレイアウト (トップページ) <div id="container">



39

見本のレイアウト (2ページ目) <div id="container">



40

marginとpadding



41

marginとpaddingの記述方法

- ◆ margin: 0; 上下左右が0
- ◆ margin: 0 auto; 上下が0 左右は中央揃え
- ◆ margin: 0 0 0 0; 上、右、下、左が0
- ◆ 個別に指定するときは
 - ◆ margin-left, margin-right, margin-top, margin-bottom
 - ◆ 例) margin-left: 100;

42

3日目

43

課題

- ◆ 9. 3つ目のHTMLファイルを作成(方法は2つ目と同じ)
- ◆ 10. W3C HTML ValidatorでHTML文法チェック
 - ◆ <http://validator.w3.org/>
 - ◆ 「Validate by File Upload」タブを選び、ファイルをアップロードしてチェック
 - ◆ エラーがなくなるまでファイルを修正
 - ◆ 注)どうしても無理ならよいです

44

課題

- ◆ 11. W3C CSS ValidatorでCSS文法チェック(トップページだけでもよい)
 - ◆ <http://jigsaw.w3.org/css-validator/>
 - ◆ 「アップロード」タグを選び、CSSファイルをアップロードしてチェック
 - ◆ エラーがなくなるまでファイルを修正
 - ◆ 注)どうしても無理ならよいです
- ◆ 12. 完成させて提出

45

法的問題

- ◆ 一般
 - ◆ **著作権関連: 他人の著作物を著作権者の許諾なしに複製してはならない、Web上に掲載してはいけない**
 - ◆ **誹謗中傷関連: 他人の名誉を傷付けたり、誹謗、中傷してはならない**
 - ◆ その他の法律違反や個人情報の扱いにも注意
- ◆ 実験室
 - ◆ 他人の権利を害する恐れがあったり、良識上好ましくない内容があると判断した場合、修正・削除を指示することがある
 - ◆ 販売活動をしてはいけない

46

課題(オプション)

- ◆ 13. cssファイルをcssディレクトリにまとめる(オプション)加算あり
- ◆ 14. 画像ファイルをimageディレクトリにまとめる(オプション)加算あり
- ◆ 15. 学内公開(オプション)加算あり

47

一般的なディレクトリ構成

- ◆ public_html(ディレクトリ)
 - ◆ css(ディレクトリ)
 - ◆ style.css(名前は自由)
 - ◆ image(ディレクトリ)
 - ◆ index.html
 - ◆ 他のHTMLファイル(あるいはフォルダ)

48

ディレクトリの整理(オプション)

- ◆ CSSファイルをcssディレクトリにまとめる
 - ◆ public_htmlの下にcssディレクトリを作成
 - ◆ CSSファイルをcssディレクトリに移動
 - ◆ HTMLのCSSファイルへのリンクのパスを変更して保存
 - ◆ 例) <link rel="stylesheet" href="style.css">を<link rel="stylesheet" href="css/style.css">に

49

ディレクトリの整理(オプション)

- ◆ 画像ファイルをimageディレクトリにまとめる
 - ◆ public_htmlの下にimageディレクトリを作成
 - ◆ 画像ファイル(.jpg, .pngなど)をimageディレクトリに移動
 - ◆ HTMLとCSSファイルの中の画像へのパスを変更して保存
 - ◆ 例) をに

50

Webにおける絶対パスと相対パス

- ◆ 絶対パス
 - ◆ ファイルのURL
 - ◆ 例) <https://www.ecs.osaka-cu.ac.jp/~学籍番号/>
 - ◆ 例) <https://www.ecs.osaka-cu.ac.jp/~学籍番号/index.html>
 - ◆ 例) <https://www.ecs.osaka-cu.ac.jp/~学籍番号/profile.html>
 - ◆ 例) <https://www.ecs.osaka-cu.ac.jp/~学籍番号/css/style.css>
 - ◆ 例) <https://www.ecs.osaka-cu.ac.jp/~学籍番号/image/logo.gif>

51

Webにおける絶対パスと相対パス

- ◆ 相対パス
 - ◆ 自分の位置から見た相手の位置
 - ◆ 自分「.」、自分の上「..」(ただし「./」は省略可能)
 - ◆ 例) index.htmlからprofile.htmlを見る場合:
 - ◆ ./profile.html または profile.html
 - ◆ 例) index.htmlからstyle.cssを見る場合:
 - ◆ ./css/style.css または css/style.css
 - ◆ 例) style.cssからphoto.gifを見る場合:
 - ◆ ../image/logo.png

52

学内公開(オプション)

- ◆ VPN接続で学内LANに入る
 - ◆ <https://intra.cii.osaka-cu.ac.jp/ocunet3vpn/>
- ◆ Proself (情報処理教育システムで使えるクラウドストレージ)を使ってファイルをアップロード
 - ◆ <https://webdav.ecs.osaka-cu.ac.jp/proself/>
 - ◆ 提出したいファイルをProselfのpublic_htmlフォルダの下にアップロード
 - ◆ 約10分後に学内の利用者からURLでアクセス可
 - ◆ URL入力で表示確認
 - ◆ <https://www.ecs.osaka-cu.ac.jp/~ログイン名/>

53

URL入力

- ◆ <https://www.ecs.osaka-cu.ac.jp/~ログイン名/>
 - ◆ 「:(コロン)」「/(スラッシュ)」「.(ドットまたはピリオド)」「-(ハイフン)」「~(チルダ)」を半角で間違えずに打つこと
 - ◆ 「;(セミコロン)」「,(コンマ)」「_(アンダーバー)」などと間違えやすい
 - ◆ 「~(チルダ)」は「0」の2つ右隣のキーをShiftキーを押しながら押す

54

学内公開後 ページが表示できないときにチェック

- ◆ファイル名
 - ◆空白を含むファイル名、日本語等のファイル名は×
- ◆画像ファイルの形式(表示できるのはPNG, JPG, GIF)
- ◆ファイルが壊れていないか
- ◆パス名
 - ◆file: では×
- ◆ファイルの置き場所
 - ◆public_htmlに下にあるか、指定された場所にあるか
- ◆パーミッションが「読めない」設定になっていないか

55

学内公開後 ページが表示できないときにチェック

- ◆文字化け
 - ◆<meta>で指定された文字コードと実際のファイルの文字コードが合っているか (UTF-8)
 - ◆ファイルを文字化け状態で保存していないか

56

提出方法

- ◆標準
 - ◆WebClassの掲示板で作成したファイル(提出したいファイル)を添付ファイルで提出
- ◆オプション(加点あり)
 - ◆学内公開後、WebClassの掲示板にURLを提出

57

HTML&CSSの経緯

- ◆1990頃 Web登場
 - ◆HTMLはWebページの論理構造を記述するもので、レイアウトを記述するものではない
 - ◆レイアウトを記述するためのスタイルシート言語は規定されなかった
- ◆1993頃以降 ブラウザ普及
 - ◆サイト製作者は見た目をよくしたい
 - ◆ブラウザにあわせてレイアウト記述
 - ◆例)文字を大きくしたいときに<h3>を使う
 - ◆ブラウザによって異なる独自要素も登場 例)
 - ◆HTMLの混乱と複雑化

58

HTML&CSSの経緯

- ◆1996 レイアウトを記述する枠組として規定(CSS1)
- ◆1997 HTML3.2:折衷的なHTML
 - ◆暫定的措置としてレイアウトに関する独自要素の取り入れ
- ◆1997 HTML4.0:論理構造とレイアウトの分離
- ◆[1998 CSS2](#)
- ◆1999 HTML 4.01
- ◆2000 XHTML 1.0
- ◆2001 XHTML 1.1

59

HTML&CSSの経緯

- ◆[2011 CSS 2.1](#)
- ◆[2011/2012 CSS3 \(CSS2.1ベースでモジュール化\)](#)
- ◆[2014 HTML5:元はWHATWGによる](#)
- ◆2016 HTML 5.01
- ◆[2017 HTML 5.02](#)
- ◆CSS4(通称)

60

CSS利用のメリット

- ◆ 詳細なレイアウトを記述できる
 - ◆ HTMLよりレイアウト記述力が高い
 - ◆ 例) テキストや画像を好きな位置に配置できる
- ◆ HTMLから、レイアウトに関する記述を除去できる
 - ◆ HTMLがシンプルに
 - ◆ 文法間違いを減らせる→情報を正しく伝達できる、表示の間違いを減らせる、維持管理が楽に
 - ◆ アクセシビリティ向上
 - ◆ 例) HTMLのtable要素を使ってレイアウトすると、音声読み上げソフトを利用してWebページを聞く視覚障害者に正しく情報が伝わりにくい
 - ◆ 検索エンジン対策

61

CSS利用のメリット

- ◆ レイアウトに関する記述を外部CSSファイルにできる
 - ◆ サイトの維持管理が楽に(後述)
- ◆ ユーザが自分の好きなスタイルシートでページを見ることができる
- ◆ 正しい(推奨される)考え方

62

HTMLとCSSの組合せ(参考)

- ◆ HTMLとCSSの組合せは自由
- ◆ HTML5 + CSS2.1(現在)
- ◆ HTML5 + CSS3(余裕のある人へ)

63

他の見本(山田太郎2)

- ◆ HTML5+CSS3
 - ◆ <https://www.ecs.osaka-cu.ac.jp/~fee26852/mihon2/>以下
 - ◆ index.html
 - ◆ css/css3.css
 - ◆ images/photo-1000-01.jpg, photo-1000-02.jpg
 - ◆ scrollssmoothly.js
 - ◆ CSS3を利用すると今風になります
 - ◆ コピーして使用してもよい

64

CSS利用の問題

- ◆ HTMLより仕様が複雑
- ◆ ユーザがスタイルシートを利用しないことがある
- ◆ ブラウザによってCSSの対応が異なる。ブラウザのバグもある
 - ◆ CSSの文法通りに記述しても正しく表示されないことがある。ブラウザの対応度合いの差はHTMLのより大きい
- ◆ 複数のブラウザで美しく正しく表示させるためには、多くの手間と時間とノウハウが必要

65

授業終了後に。。。

- ◆ 今期は遠隔授業のため学外公開ができません
- ◆ 学外公開用パーミッションの設定は教室のパソコンでないとできないため
- ◆ 授業終了後にチャレンジしてみてください

66

パーミッション

- ◆ディレクトリ(フォルダ)やファイルのアクセス権
 - ◆d: directory; r: readable; w: writable; x: executable

d	r	w	x	r	w	x	r	w	x
	Owner			Group			Other		
-	r	w	-	r	-	-	r	-	-

(例) このファイルは
自分(オーナー)は読み書き可能
他人は読むことができるだけ

67

パーミッション

- ◆他人に公開
 - ◆フォルダ、ファイルのプロパティ
 - ◆Other:「読み取りと実行」を可能とする

68

Webで情報を探すためには

- ◆求める情報のURLがわかればよい しかし
- ◆情報のURLは多くの場合わからない そこで
- ◆さまざまな検索サービスがうまれた
- ◆検索エンジン(Webページを収集し、ページ内のキーワードとURLを結び付けるデータベース)
 - ◆Google, Yahoo!, Bingなど
- ◆学外公開すると、作成したページを検索エンジンで見つけてもらえるかもしれません!

69