

The background of the slide is a light gray surface covered with a dense layer of small, five-pointed star-shaped confetti. The confetti is in various colors, including gold, silver, and a pale blue. A single, thin, gold-colored wire is draped across the scene, starting from the bottom left, looping around a silver paperclip in the center, and extending towards the top right corner.

Webデザイン実習4B

2019/12/10

Kazuma Sekiguchi
class@cieds.jp

AdobeMAXJapan2019

- ぶっちゃけどーでした？



- 5300名だって。来年は11月24日（火）らしい

課題

- スマートフォンに最適化したウェブサイトの構築
 - RWDにする必要は無い（してももちろんOK）
- レストランの紹介および予約サイト
 - 名前や場所などは適当に
 - 画像は商用利用可のもの、または自分で撮影、作成したもののみ許可
 - キャラクターは自分で作成したもの以外使用禁止
- トップページ、紹介部分、予約ページが必要
- 効果的にインタラクションを用いること

課題

- 幅414px 高さ896pxのiPhone11をターゲット
 - Chromeのモバイルモードで表示可能であること
 - 確認はChrome78のモバイルモードで確認します
- jQueryはバージョン3以上のみ使用許可
 - 動かないプラグインが多数出てくるので、注意すること
- 動画の利用はもちろん可
- 各ページにリンクを張ること
 - 1ページサイトもOK。ただし、ページ内リンクは用意すること
- 12月17日の授業時に提出

課題提出物

- HTML
- CSS（HTML埋め込みでもOK）
- JS（HTML埋め込みでもOK）
- 画像ファイル群、動画ファイル群
- デザインファイル（psd , ai , sketch , xd , figmaなどなど）、バージョンは最新版でOK
 - デザインファイルのPDFファイル

<html5 APIs Canvas>

Canvas

- Canvas自体はHTML5で登場したタグ
 - ドキュメント内に絵を描くための場所を生成するタグ
- JSを利用することで、Canvasに対して任意の画像を作成することが可能
 - JSを使って絵を描く、ということ
 - アニメーションも可能
 - イメージ画像をCanvas上に出力することも可能
 - 画像の上にJSで描いた画像を載せる、ということも可能

Canvasの使い所

- プログラミングで画像を生成、制御できるため、ゲームの作成や高度な表現に利用される
 - Web上で3D表現を行うWebGLもCanvasの延長
 - 画像の加工なども可能
 - アニメーションの作成
- オンラインゲームもほとんどがCanvasを利用
 - 比較的高速な描画が可能（ある程度工夫は必要）
- AnimateCCから書き出すことが可能

Canvasのデフォルト

- 結構使いづらい
 - JSを駆使して記述が必要
 - ブラウザーごとの差異が比較的大きい（特に小数点以下の扱いが異なる。サブピクセルレンダー。Chromeが結構ダメ）
 - レイヤー機能などは存在しない（自分で作成するか、外部ライブラリを使うのが一般的）
 - 描いたものはすべて同一のものとして扱われる
 - ペイントソフトと同じ扱い
 - 異なるものを描いたとしても同一扱いのため、それぞれに違うことをすることが不可能

ライブラリ

- jQueryをベースにしたjCanvasというのもあるが . . .

- PixiJS

<http://www.pixijs.com/>

WebGLベースで動作するため高速

- EaselJS

<https://www.createjs.com/easeljs>

AnimateCCから書き出すときに使用される

- p5.js

<https://p5js.org/>

インタラクショナルアート向き

Canvas

- Canvasタグを利用する際は、width値とheight値、id名を付与する
 - width値とheight値はそのままCanvasの大きさになる
 - width属性とheight属性を記入しないとブラウザによってはまともに表示されない
 - RWDで利用する場合は、JSで動的に画面サイズに合わせてwidth属性の値を変更するなどの対応が必要
 - id名はJSからcanvasを識別するために必要
 - canvas自体は複数設置が可能
- JSからcanvasを利用する場合は、id名をキーにしてCanvasオブジェクトを取得し、そこに対して、描画を行っていく

図形の描画

- 矩形
 - `ctx.fillRect(x, y, width, height)`
 - 塗りつぶした矩形の描画
 - `x,y`：座標位置、`width`：幅、`height`:高さ
 - `ctx.strokeRect(x, y, width, height)`
 - 枠線だけの矩形の描画
- 円
 - `ctx.arc(x, y, radius, startAngle, endAngle, anticlockwise)`
 - 円や円弧の描画
 - `x,y`：座標位置（中心）
 - `radius`：半径
 - `startAngle`：円弧を書き始める角度（ラジアン指定）
 - `endAngle`：円弧を書き終える角度（ラジアン指定）
 - `anticlockwise`：円弧を描く向きを真偽値で指定。`true`で反時計周り
 - ラジアン：角度*Math.PI/180

色の指定

- `ctx.strokeStyle = color`
 - 枠線の色
 - `color`はCSS色指定が可能（rgba指定で半透明も可能）
- `ctx.fillStyle = color`
 - 塗りつぶしの色
- グラデーションの指定の仕方は異なる

画像の読み込み

- 任意の画像ファイルを読み込み可能
 - ブラウザーが扱うことができるファイルであることは最低条件
 - 実のところ映像も扱う事が可能
 - iOSで動画再生がインラインでできなかったときに代用された
- `ctx.drawImage(“ファイル名” ,貼り付ける場所 (X座標) ,貼り付ける場所 (Y座標) ,貼り付けるサイズ (幅) 、貼り付けるサイズ (高さ));`
- 縮小したり拡大して貼り付けることが可能
 - 貼り付けた後は、普通のピクセルデータの扱いになる