

ポートフォリオ

サミュエル ウィリアム ボールズ
フロントエンドエンジニア



<https://samuel-bowles.github.io/jp/index.html>

自己紹介

サミュエルボーラズと言います。イギリス出身です。ウェブ開発のフロントエンドエンジニアを目指しています。日本のいろいろなところに住んで、もう十年以上になりますが、現在は東京の目黒区に住んでいます。最初の七年間は東京で英語の教師をして、その後はITサポートの仕事をしました。その他に、北海道のスキー場の隣にあるアウトドアショップで働いたこともあります。その間に現地の山と雪そして日本の自然にすっかり夢中になって、冬にはスノーボード、夏にはハイキングをよくしました。そのため、私のポートフォリオにはスノーボードのテーマがよく出てきます。私の趣味はスノーボードとゲームですが、それ以外の時間はJavaScriptでシンセサイザーを作る方法を学ぼうとしています。ただ、これはプログラミングを学ぶというより、サウンドエンジニアリングを学ばないといけない事に最近気づきましたが、楽しいので熱心になって取り組んでいます。



スキル

開発

HTML5

HTML構文を深く理解することで、適切に構成され、構造化されたセマンティックマークアップを作成することができます。

CSS3

私は、一般的なタグと、キーフレームアニメーションなどの、より高度ないいくつかの機能を使用することができます。深い知識を持っています。複数の画面サイズに適応するレスポンシブサイトを作成できます。

JavaScript

DOMと相互作用して動作する独自のスクリプトを作成することができます。また、GoogleマップなどFirebaseデータベースなどのAPIを使用することと、外部のモジュールを使用してページのインタラクティブを高めることもできます。

jQuery

jQueryを使用して、アニメーションや視覚効果を作成したり、スライドショーや画像拡大鏡などのページコンポーネントを実装したりできます。オンラインソースのプラグインを使用して、よりインタラクティブなエクスペリエンスを作成できます。

Bootstrap 5

Bootstrapを使用してページレイアウトを作成し、スライドショーやメニューなどのBootstrapコンポーネントをWebページに挿入できます。

ツール

Adobe Photoshop

写真の調整や合成画像の作成ができます。ウェブサイトで使用するバナーを作成できます。画像のサイズを変更し、Web上のさまざまな状況で使うための正しいファイル拡張子を使用できます。

Adobe Illustrator

ロゴ、バナー、Webページ全体のデザイン構成を作成できます。Illustratorを使用して、Webページで直接使用するスライスを作成したり、Webで使う画像のサイズを変更したり調整したりできます。

Wordpress

Wordpressを使用して、ブログスタイルのWebページを作成し、オンラインソースからテンプレートテーマをインストールして調整できます。

Git/Github

私はGitの基本機能を使用する知識と能力を持っており、Githubをオンラインおよびオフラインで使用してプロジェクトやWebページをホストすることができます。

Google Firebase

データストレージ用の基本的なデータベースを作成し、JavaScriptとFirebase APIを介してそこに保存されているデータにアクセスして操作できます。

言語

英語: 母国語

日本語: 大学の学位, N2

フランス語: 大学の学位

ブラックホーク スノーボード

早稲田電子学校の個人プロジェクト

時間: 10日間 - 素材集め: 2日間、 ワイヤーフレーム: 2日間、
デザイン: 2日間、 コーディング: 4日間。

コンテンツ: トップページ:ブランド紹介。 商品ページ:商品詳細と写真。 店舗ページ:店とスタッフ紹介。 ニュースページ:ニュースと今後のイベント。 アクセスページ:店の所在地、アクセス説明、コンタクト。

制作概要: 観光地で新しいブランドを作り、新しい店舗を設立するというクライアントからのリクエスト。ウェブサイトの目標は、ブランド、製品、新しい店舗の場所の認知度を高めるためのオンラインプレゼンスを設けることです。さらに、この店舗はアクセスしにくい場所にあるため、顧客が実店舗を見つけるのに役立つはずです。

技術: HTML, CSS, JavaScript, jQuery.

このプロジェクトで3つの主要な目標を設定しました。1つ目は、プララックス効果を使用して、印象的なトップページのデザインを作成することでした。2つ目は、製品ページを面白くて魅力的な方法で表示することでした。3つ目は、国際的な顧客ベースが期待されるため、多言語サイトを提供することでした。

ツール: VS Code とライブサーバープラグイン , Notepad++, Inkscape, Illustrator, Photoshop, Blender



<https://samuel-bowles.github.io/blackhawk-snowboards/>



Photoshop:
画像あわせとマスキング.



Inkscape:
画像 .svgファイル変化
パス作成、着色



このサイトのデザインはロゴの作成から始まりました。ブランド名を出発点として、InkscapeとPhotoshopを組み合わせてこの.svgロゴを作成しました。

サイトのデザインスキームはロゴをベースにしたシンプルなカラーパレットを使用して、シンプルでありながらスタイリッシュな効果を出しました。

スノーボードの輪郭で結ばれた3枚の画像に視差効果を使ってメインページのデザインを作成しました。これは、「background-attachment:fixed」プロパティを介してCSSのみで実現しました。

ただし、このプロパティはiOSで完全にはサポートされていません。その結果、ページレスポンシブレイアウトは異なるデザインを使用しています。



background-attachment: fixed

視差スクロール

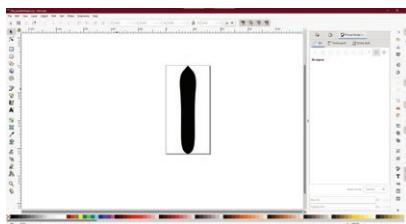


視差スクロール



スノーボード 商品ページ

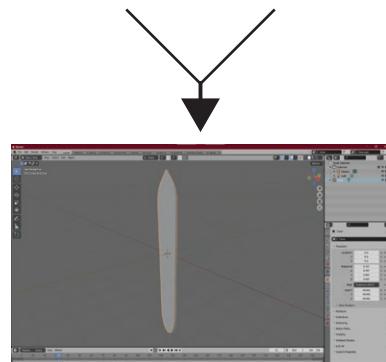
スノーボードはこの店の主力商品であり、実際にどのように見えるかをユーザーが実感できるように見せたかったのです。大きな画像は、ユーザーがデザインと製品の詳細を見るのに役立ちます。ただし、静止画像を見るだけでなく、製品をより具体的にするために、スノーボードの3Dモデルをレンダリングすることにしました。これにより、ユーザーはより触覚的な方法で製品を操作し、理解することができます。



Inkscapeで.svg 輪郭



Adobe Illustratorデザイン



ブレンダーで3D モデルとマスキング。
.svgの押し出しと.pngマスキングの合わせ

ファイル名拡張子は.gltfで、軽量で、
クロスブラウザでサポートされているWebファイル形式です。

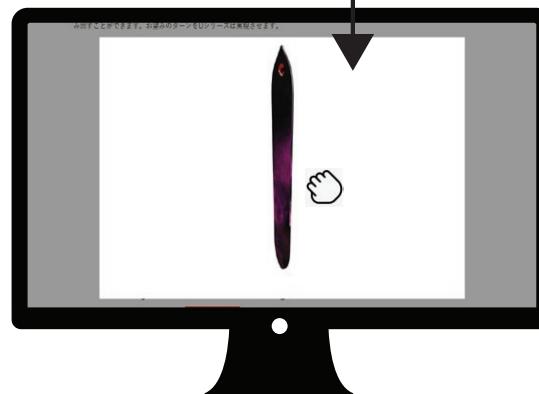
3D ビューアー実装

ユーザーは、興味があるスノーボードまでスクロールすることで、ボードのデザインの画像と詳細を調べることができます。必要に応じて、[3Dビュー]ボタンをクリックして、製品のインタラクティブな3Dビューを開くことができます。



このボードは、あらゆる条件の山全体をサーフィンできるように設計されています。グルーマーではその長い効果的なエッジが、優れた安定性とコントロールを与え、軽いスイングウェイトとテールフレックスにより、美しい弧を描き、脱く優雅に曲がることができます。パウダースノーではどのように機能するか説明する必要はありません。ストラップを付けて、その心地よさを体験してください。

各スノーボード詳細部分にある「3Dビュー」ボタンをクリックすることができます。



Googleの提供している3D モデルビューアーはGithubにあります。
<https://github.com/google/model-viewer>

ビューアーをボタンクリック表示するためにjQueryのモーダルウインドウ挿入スクリプトを書いて、フェードの効果もつけました。
このソリューションはデスクトップもモバイルでも使用することができます。

店舗ページ

ショップページでは、ショップやスタッフを紹介するために高品質の画像を用意することが重要でした。これにより、ページサイズとページの読み込み時間が長くなります。これを相殺するために、jQueryを使用する代わりに、ショップの画像用にJavaScriptで単純な画像スイッチャーを書きました。次に、ユーザーがブラウザでJavaScriptを無効にしている場合でも機能するCSSのみの3Dカードビューアーを作成しました。

大きなイメージを見るためにクリック

The screenshot shows a navigation bar with links: home, snowboards, shop (highlighted), news, access, ENGLISH. Below the navigation is a section titled 'Shop' with a grid of five small images. One image in the top row is larger than the others. A callout bubble points to this larger image with the text '大きなイメージを見るためにクリック' (Click to view the large image). The large image shows a snowy street with buildings.

当店は現在オープンしております。いつでもお立ち寄りください。店内のカフェでコーヒーをお楽しみください。

Staff

フレンドリーなスタッフが、豊富な専門知識と、美味しいコーヒーでおもてなしします。

A grid of three staff portraits: Joe Sato, Vika Henderson, and Joe Bliggo. Each portrait has a callout bubble pointing to it with the text '大きなイメージを見るためにクリック'.

Akira Sato is 31 years old, born and raised in a small town of Japan. He grew up snowboarding in a small DIY park on a tiny hill behind his school in Poland. It wasn't even a hit, it was basically just a pile of snow. Joe is a snowboarder who loves to ride and has won several awards in his career. He currently works at Black Sheep Snowboards as a shop manager and enjoys spending time with his family and friends.

Vika Henderson is a Polish snowboarder who started snowboarding in a small park on a hill behind her school in Poland. She quickly became one of the best snowboarders in her country and won several national titles. She now lives in the US and works as a shop manager at Black Sheep Snowboards. In her free time, she enjoys snowboarding and traveling.

Joe Bliggo is a snowboarder from the US, born on 2/23/1989. He graduated with a degree in business but promptly ignored that to follow his passion for snowboarding. He currently works as a shop manager at Black Sheep Snowboards and enjoys spending time with his family and friends.

home | snowboards | shop | news | access | ENGLISH

Copyright © 2022 Black Sheep Snowboards

```
thumbChanger.js
1 document.addEventListener("DOMContentLoaded", function() {
2     let thumbs = document.querySelectorAll(".shop-small");
3     let bigPic = document.querySelector(".shop-large");
4
5     let captionsJp = ["当店の外観と入り口.", "「ゆっくりして楽しむ」がモットー。", "冬の夜の大通り。", "リゾートビレッジの雪上からの風景。"];
6     let captionsEn = ["The main entrance of our store.", "Enjoy a relaxing coffee.", "The main street at night.", "View of the resort village from the snow."];
7     let lang = document.getElementsByName('lang')[0].getAttribute('lang');
8
9     thumbs.forEach(element) {
10         element.addEventListener('click', function(event) {
11             for (let i = 0; i < thumbs.length; i++) {
12                 thumbs[i].classList.remove('selected');
13             }
14             this.classList.add('selected');
15
16             let captions;
17             if (lang === 'ja') {
18                 captions = captionsJp;
19             } else if (lang === 'en') {
20                 captions = captionsEn;
21             } else {
22                 return false;
23             }
24
25             let clickedElemId = event.target.id;
26             changeSource(clickedElemId, bigPic, captions);
27         })
28     }
29
30     function changeSource(clickedElemId, bigPicElem, captions) {
31         clickedElemId = clickedElemId.slice(0, -5) + clickedElemId;
32         let newSrc = bigPicElem.src.slice(0, -5) + clickedElemId + ".jpg";
33         bigPicElem.src = newSrc;
34
35         let figCap = document.querySelector('.large-figcap');
36         figCap.innerText = captions[clickedElemId - 1];
37     }
38 });
39
40 });
41 
```

イメージはJavaScriptで交換する



ユーザーはいつでも日本語と英語で言語を変更することができます。figcaptionも翻訳され、JavaScriptを使用してページに挿入されます。

CSS::before タップして反転



ショップスタッフを紹介するCSSのみのカードフリッパーを実装しました。CSS transformとbackface-visibilityプロパティは、この特定の効果を可能にしました。ただし、モバイルでのホバーは発見しづらいため、CSSのみでモバイルに表示される小さなサインも追加しました。

Glampia

早稲田電子学校 グループプロジェクト

役割: コーダー

時間: 10日間、 1日クライアント相談、 1日素材集め、 4日間 フィーチャーテストとワイヤーフレーム、 3日間コーディング、 1日微調整

コンテンツ: トップページとグランピング紹介、 部屋ページ、 食事ページ、 アクセスページ、 コンタクトページ。

制作概要: クライアントからは、グランピングビジネスを宣伝するためのウェブサイト作成の要求が来ました。このサイトの主な目的は、この活動に対する認識とターゲットの関心を高めるためです。その他、新しいグランピングのキャンプ場に関するアクセス情報を顧客に提供することでした。

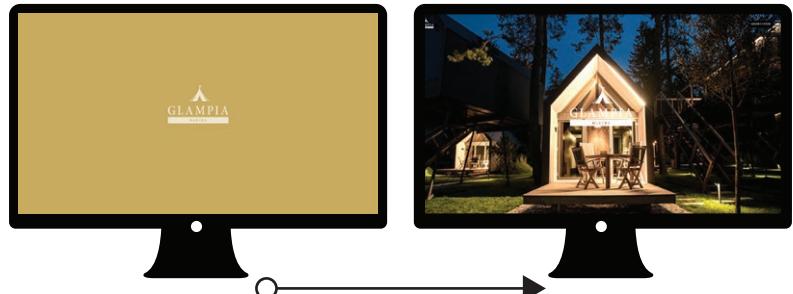
技術: HTML, CSS, JavaScript, jQuery.

私たちの目標は、これらのテクノロジーを使用して、グランピング体験を最もよく見せるための強力なビジュアルサイトを作成することでした。リードコーダーとして、jQueryのアニメーション機能を使用して、ユーザーに快適なエクスペリエンスを提供するサイトを作成したいと考えました。

ツール: Sakura editor, Notepad++, Illustrator, Photoshop.



<https://samuel-bowles.github.io/glampia/>



jQueryとJavaScriptを使用してスプラッシュスクリーンからスライドショーにフェード

ユーザーが最初にサイトにアクセスすると、会社のロゴがスプラッシュスクリーンに表示され、キャンプサイトの画像を含むスライドショーにフェードインします。これを実現するために、スプラッシュスクリーンのCSSクラスをフェードインおよびフェードアウトするように変更するjQueryスクリプトを作成しました。また、サイトの使いやすさを損なう可能性があるため、ユーザーがページにアクセスするたびに画面が表示されないようにするトラッカーをローカルストレージに含めました。

```
splashPage.js [O]
1 $(document).ready(function() {
2
3     function loadDb() {
4         return JSON.parse(localStorage.getItem('splashCount'));
5     }
6
7     function populateStorage(sC) {
8         localStorage.setItem('splashCount', JSON.stringify(sC));
9     }
10
11    let splashCount = loadDb() ? loadDb() : 0;
12    splashCount += 1;
13
14    if (splashCount == 1) {
15        $("header").addClass("header-hidden");
16        $("header").append("<div class='splash'></div>");
17        $(".wrapper").css({"opacity": 1});
18        $(".splash-logo").css({"z-index": 99});
19        $(".splash-logo").fadeTo(2000, 1).fadeTo(500, 0, function() {
20            $(".splash-logo").css({"opacity": 1, "z-index": 1, "position": "relative"});
21            $(".splash").fadeOut(500);
22        });
23        populateStorage(splashCount);
24    } else if (splashCount == 5) {
25        splashCount = 0;
26        populateStorage(splashCount);
27    } else {
28        populateStorage(splashCount);
29    }
30
31});
```



jQuery

ページ上のHTML要素にフェードイン効果を実装するために、特定のHTML要素がビューポートのビュー内にあるかどうかを計算するinview(<https://github.com/protonet/jquery.inview>)と呼ばれるプラグインを使用しました。CSSクラスを追加または削除して、Y軸上の要素を変換し、不透明度を変更するスクリプトを書きました。

スクリプトとCSSクラスを設定して、HTML要素にfadeクラスを適用するだけで効果を任意の要素に適用できるようにしました。スクリプトとCSSをこのように設定すると、JavaScriptが無効になっている場合でも、ページにすべての情報が表示されます。

```
$(function() {
    $( document ).ready(function() {
        $(".fade").addClass("fade-hide");
    })

    $(".fade").one("inview", function(event, isInView) {
        if (isInView) {
            var mfThis = $(this);
            mfThis.addClass("fade-in");
            setTimeout(function(){
                mfThis.removeClass("fade-hide fade-in");
            }, 1100 );
        }
    });
});
```

```
/* Add fade in on scroll */
.fade {
    opacity: 1;
    transform: translateY(0);
}

.fade-hide {
    opacity: 0;
    transform: translateY(50px);
}

.fade-in {
    opacity: 1;
    transform: translateY(0);
    transition: 1s;
}
```

私たちのアクセスページはもともと静止画像のルートマップを使用していました。ただし、スクリーンリーダーや検索エンジンは、その情報を読み込むことができません。したがって、HTMLとCSSのみで完全にレスポンシブでアクセス可能なバージョンを実装しました。テキストの方向も変わり、モバイルディスプレイでの読みやすさが向上します。



メニューバーのビジュアリティ

もう1つの重要なユーザビリティの側面は、トップメニューの可視性でした。チームのデザイナーは、各ページの上部にあるメインビジュアルで強いインパクトを与えることを想定して、上部のメニューを非表示にする必要がありました。ただし、フォントとロゴは白であるため、ユーザーがページを下にスクロールすると「非表示」になる可能性があります。解決するために、ユーザーがメインビジュアルを超えてスクロールしたときに、トップメニューの背景色を塗りつぶすJavaScriptのスクリプトを書きました。メインメニューでは使いやすさが重要なので、JavaScriptが無効になっている場合にメニューに色を付けるCSSフォールバックソリューションも含めました。

メニューバー非表示



JavaScript オン

メニューバー表示



JavaScript オン

メニューバー表示



JavaScript オフ

メニューバー表示



JavaScript オフ

```
.header-hidden {
    background: rgba(0, 0, 0, 0);
}

/* Add transparent background to header on scroll (jquery trigger) */
.fillBg {
    background: linear-gradient(rgb(0, 0, 0, 0.6), transparent);
    animation-name: fadeInBg;
}

@keyframes fadeInBg {
    from {
        background: transparent;
    }
    to {
        background: linear-gradient(rgb(0, 0, 0, 0.6), transparent);
    }
}
```

CSSキーフレームアニメーションは、JavaScriptによって作動されると背景色を変更します。

JavaScriptは、ウィンドウの上部からスクロール位置を計算します。画像の高さがわかっているので、ユーザーが画像を通り過ぎる前にエフェクトを作動して、フェードインが目立たないようにすることができます。

```
$function() {

    $("header").addClass("header-hidden");
    let page = document.body.id;

    if (page == "top-page") {
        $(window).scroll(function() {
            if ($(document).scrollTop() > $(window).height() * 0.9) {
                $("header").addClass("fillBg");
            } else {
                $("header").removeClass("fillBg");
            }
        })
    } else {
        $(window).scroll(function() {
            if ($(document).scrollTop() > 300) {
                $("header").addClass("fillBg");
            } else {
                $("header").removeClass("fillBg");
            }
        })
    }
});
```

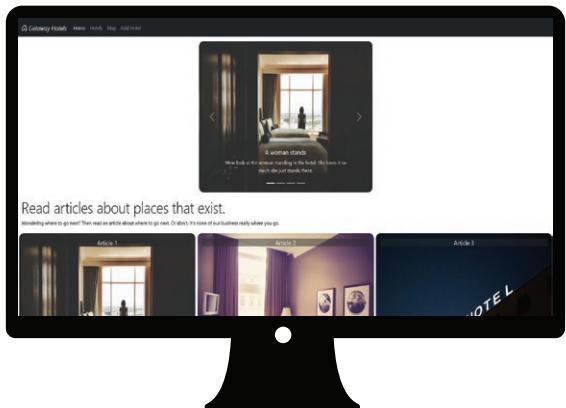
ホテル

個人プロジェクト

時間: 15 日間

目的: Bootstrap、Google FirebaseとGoogle Maps APIを含めて新し技術を習うため。

技術: HTML, CSS, JavaScript, Bootstrap 5, Google Firebase, Google Maps API, Stripe Payments



<https://samuel-bowles.github.io/hotel-website/>

The mobile version of the website features a header with the 'Gateway Hotels' logo and navigation links. The main content area includes a large image of a woman in a hotel room, a grid of nine article cards labeled Article 1 through Article 9, and a footer with social media icons and copyright information.

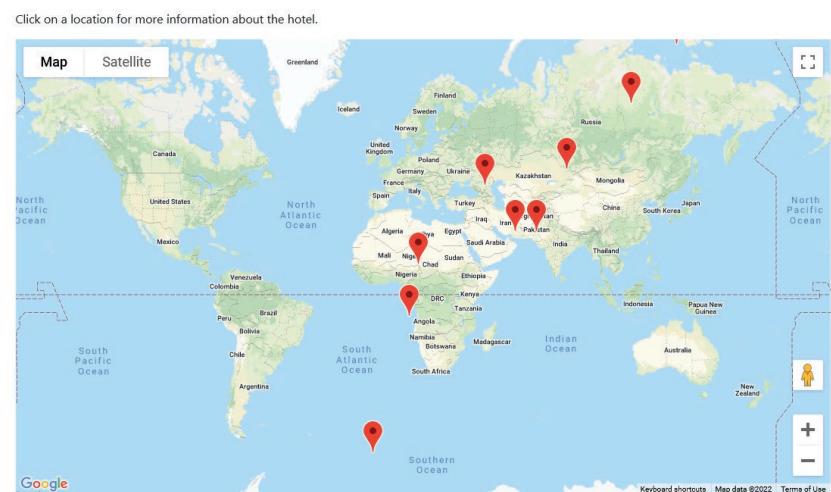
Bootstrapを使用して、レスポンシブとハンバーガースタイルのメニューを含むグリッドのようなレイアウトをすばやく実装することができました。

Cloud Firestore location: asia-northeast1

ホテルに関するデータをホストするためにGoogleFirebaseを使用しました。Javascriptを使用してAPIを介してフェッチできます。データベースに簡単にアイテムを追加できるようなフォームも書きました。

JavaScriptは、FirebaseAPIを使用してホテルのデータを取得します。次に、作成したBootstrapテンプレートを使用してページにデータを入力します。ユーザーは、さまざまなアメニティタグを選択することで、クライアント側でホテルをフィルタリングできます。

「今すぐ予約」をクリックして



ホテルのデータセットとGoogle MapsAPIを使用してGoogleMaps上のホテルのロケーションマップを表示することができます。これにより、ユーザーは場所ごとにホテルを選択でき、地図アイコンをクリックして各ホテルに関する情報を表示することもできます。

Stripe支払い処理ページが表示されます。

コードスニペット

ここでは、テスト目的で作成したパートのコードスニペットをいくつか含めていますが、最終的には完成したWebページにはなりませんでした。

JavaScript カレンダー

2022年3月						
Sun	Mon	Tue	Wed	Thr	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

これは、Dateオブジェクトを使用して作成された単純なJavaScriptカレンダーです。ユーザーは、カレンダーを前後にクリックしたり、現在の月に戻したりすることができます。

ここでは、各面に個別の画像を含むCSSのみの自動回転カードを作成しました。残念ながら、この特定の実装は応答しないため、Blackhawk Snowboards Webサイトで変更されたバージョンを使う必要がありました。



jQuery画像引伸機

この画像拡大表示は、神田幸恵の本に記載されていた「jQuery標準デザインレッスン」のコードに基づいていましたが、本の固定レイアウトではなく、レスポンシブデザイン内での機能するようにコードを拡張しました。

