情報システム開発基礎演習

ITインフラ編 第01回

3年、春学期、必修

オリエンテーションのお品書き

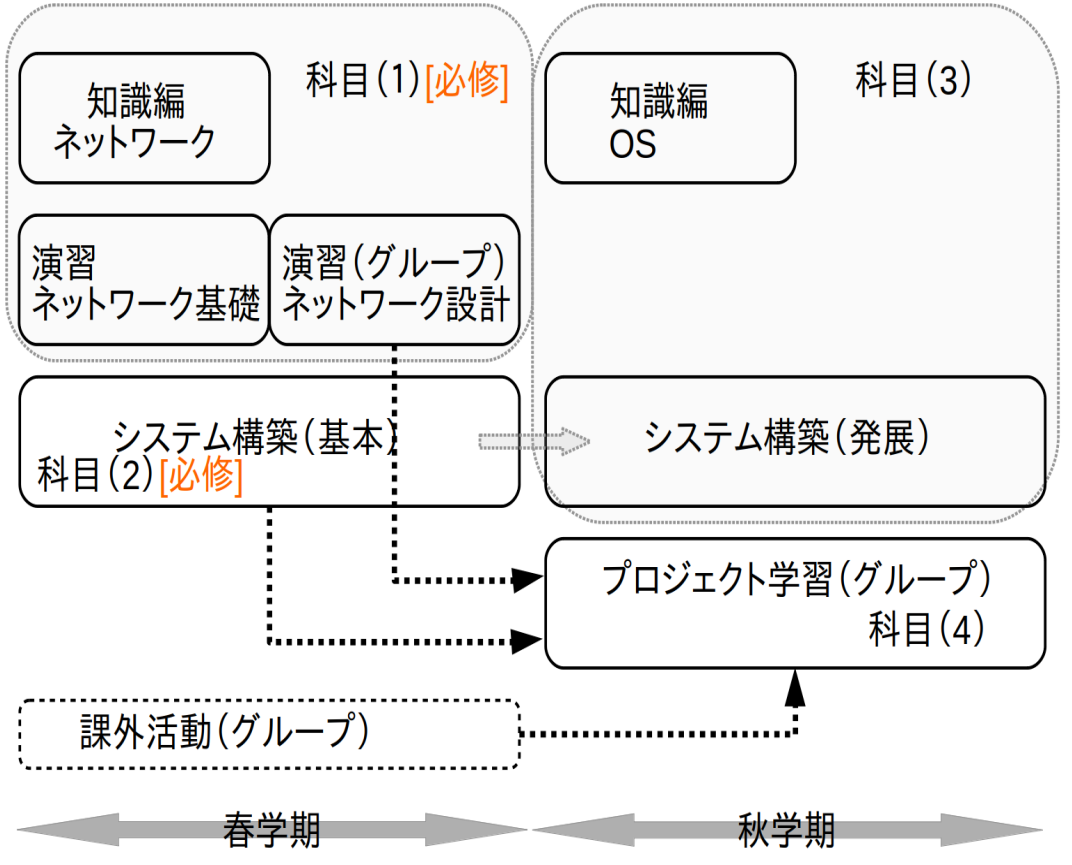
1. なぜITインフラが分かった方がいいのか？(私論)
2. ITインフラ演習科目群における本科目の位置づけ
3. 本科目の詳細
    - 目的,目標
    - シリーズ構成
4. 演習のイメージを実演・紹介
5. 用語
6. 目標設定

なぜITインフラが分かった方がいいのか？(私論)

- フレームワークとかSDKとか便利なものを使えば、さくっと色々なアプリを作れはするのですが、 いったい何がどう動いているのか分からない（-> モヤモヤする）
- モヤモヤするならフタを開けて裏側をのぞいてみましょう。 案外かんたんな理屈です
- フタを開けると、
  - 動作原理（理屈）が分かります
  - 原理が分かれば、エラーメッセージから原因が類推できるようになります
  - ほかの技術も似たり寄ったりなので、新技術の目利きも出来るようになります
  - そして、なにより楽しいじゃないですか、フタを開けるのって
- 10年、20年先のキャリアにつながることがあります
  (楽しいことで御飯が食べられたら最高ですよね?:-)
  - (1)究める(2)オールラウンダーはなかなか厳しいけど得意分野ITインフラはどう?

ITインフラ演習科目群における本科目の位置づけ

- 科目(1) ... 「コンピュータネットワーク」
  月曜午前
- 科目(2) ... 「情報システム開発基礎演習」のITインフラ編(01回〜07回の前半);月曜午後
- 科目(3) ... 「クラウドコンピューティング」(秋)
  - 科目(2)は体験でしかないので、秋学期は、もうすこし本気のシステム構築をやります
- 備考
  - 科目(1)(3)に「知識編」とありますが、演習に必要な知識を少し(スライド数枚)講義するだけです。 これらの授業は基本情報処理試験の範囲ですら全然カバーしていません。 試験を受ける人は各自でがんばってください



(脚注) 基本情報処理試験の範囲ですら内容が多すぎるのだそうで、その授業は、やらなくなりました

目的・目標

目的

- クラウドを使ったシステム構築の体験

目標

1. クラウド(AWS)を少し経験する
2. 基本的なサーバが構築できる
3. AWSサービスとの連携が出来る

シリーズ構成: 概要(#01-07前半)

1. サーバを作成できる(#02)
2. WWWサーバをたてられる(#03-05)
3. AWS Rekognitionと連携できる(#06)

ただし、経験するだけなので、必須課題は、ほぼノーコードシステム開発です ;-) 余力がある方は自由課題にレッツトライ！

本日の後半はAWS演習

- では、少し休憩したら、AWSを体験してみましょう
  - 招待したコースがうまく動いてないみたい?いま再検討中なので、少々お待ちを
- 演習
  1. AWSにログインする
  2. AWS Consoleに入る
  3. AWSの色々なサービスをさわってみる

用語

- クラウド (cloud)
  - この授業での定義は「非常に多くの在庫を持つ仮想基盤がクラウド」
  - 在庫が豊富なので「今すぐサーバを1000台起動したい」といったことが可能な脅威のサービス
  - 商品名、サービス名。定義が不明なバズワード。Eric Schmidtが2006年の講演でcloudと言った説
- AWS (発音：えーだぶりゅーえす) ... Amazon Web Services
  - もっとも有名な事実上の業界標準であるクラウドサービス
- Unixオペレーティングシステム (発音：ゆにっくす)
  - このテキストではDebian GNU/Linuxのことです
    (初心者は50年以上にわたるUNIX開発の複雑な系統図なぞ気にしなくてOK)
  - 世間的には「黒い画面にコマンドを打ちこむ謎の黒魔術」的な何かと思われていそうです;-)
- Linux　（発音：りなっくす、りぬ〜くす）
  - クラウドサービスでもっともよく使われるOS
  - UNIXのニセモノ(Unix クローン)のひとつ

本日の課題

1. AWSのサービスは色々あります。あれこれ、さわってみてください

2. ポータルの授業ページで「目標設定」をしてください。

    - 本日は、これで課題提出とみなします
    - 目標設定のその他欄(あるんだっけ?そんなの)に、AWSの感想も書いてくれていいのよ
    - 最終課題で、こういう連携やってみたいな～とか

- AWS Academyで遊べる例
    - 例: AWS Rekognitionに画像をアップロードすると、判定して、ラベル付け
    - 例: Amazon Textractは「印刷されたテキスト、手書きの文字、レイアウト要素、データを、あらゆるドキュメントから自動的に抽出」
    - 例: Amazon Lex は、アプリケーションに会話型インターフェイスを設計、構築、テスト、およびデプロイするための高度な自然言語モデルを備えた、フルマネージド型人工知能 (AI) サービス

(脚注) (1)「はぁ～クラウドっぅやつは便利じゃのぉ～」みたいな？ (2)ニューラルネットワークを書かなくてもいいんですね～(棒読み)

情報システム開発基礎演習

ITインフラ編 第02回

3年、春学期、必修

本日の目標

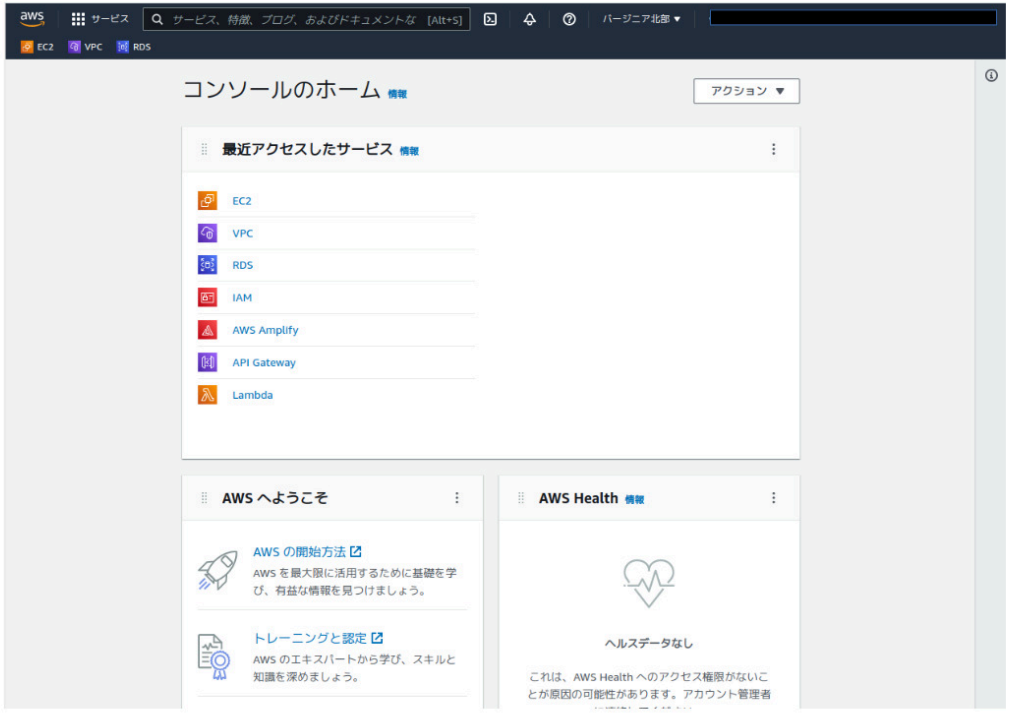
1. AWSをつかう体験
2. AWSでサーバを構築できる
3. (来週以降に使う)演習支援環境のセットアップ

お品書き

1. AWS Academyへのログイン(復習)
   - みなさんAWS(本物)の管理画面(AWS Console)は出せていますね？
2. おしらせ
   - Unixコマンドの同人誌くばってます(近日開始予定)。電子版は、こちらからどうぞ
   - 秋には、第2版「とどけたいコマンド」を製作予定(なぞ)
3. 用語
4. EC2の作成
5. システム構築演習支援システム(Colitas)のインストール
6. Elastic IP (EIP)の取得
7. TA/SAさんに確認してもらい終了
   - システム構築演習支援システムのインストールが出来ていることを確認してもらってください
   - EIPを所定のフォーム(google form)に入力したことを確認

この画面が出せていますね？

- (画面の詳細は異なると思いますけど) これっぽい画面が出て入ればOK
- AWS Academy (vocareum)の段階で、つまづいている人は、 (ポータルの第2回添付資料にある)先週分の作業のマニュアルを見てください。 ちなみに動画もあります
- これをAWS Consoleと呼んでいます。 ようするにAWSの管理画面のことです
- ではサーバ構築に進みましょう



用語

- プロトコル ... Protocol
  - 相互に取り決めた約束事。インターネットの場合は「通信規約」といった意味
- IPアドレス(発音：あいぴーあどれす) ... IP Address
  - コンピュータを識別する数字。 ちなみにIPはInternet Protocolの略
- コマンド
  - コンピュータに与える命令で擬似英語っぽい単語(+文章)、スペース区切りに注意
- AWS (発音：えーだぶりゅーえす) ... Amazon Web Services
  - もっとも有名な事実上の業界標準であるクラウドサービス
- Unixオペレーティングシステム (発音：ゆにっくす)
  - このテキストではDebian GNU/Linuxのことです
- Linux　（発音：りなっくす、りぬ〜くす）
  - クラウドサービスでもっともよく使われるOS
  - UNIXのニセモノ(Unix クローン)のひとつ

(脚注) 先週分および午前中の授業と一部重複しています。午前の復習も兼ねているわけですね

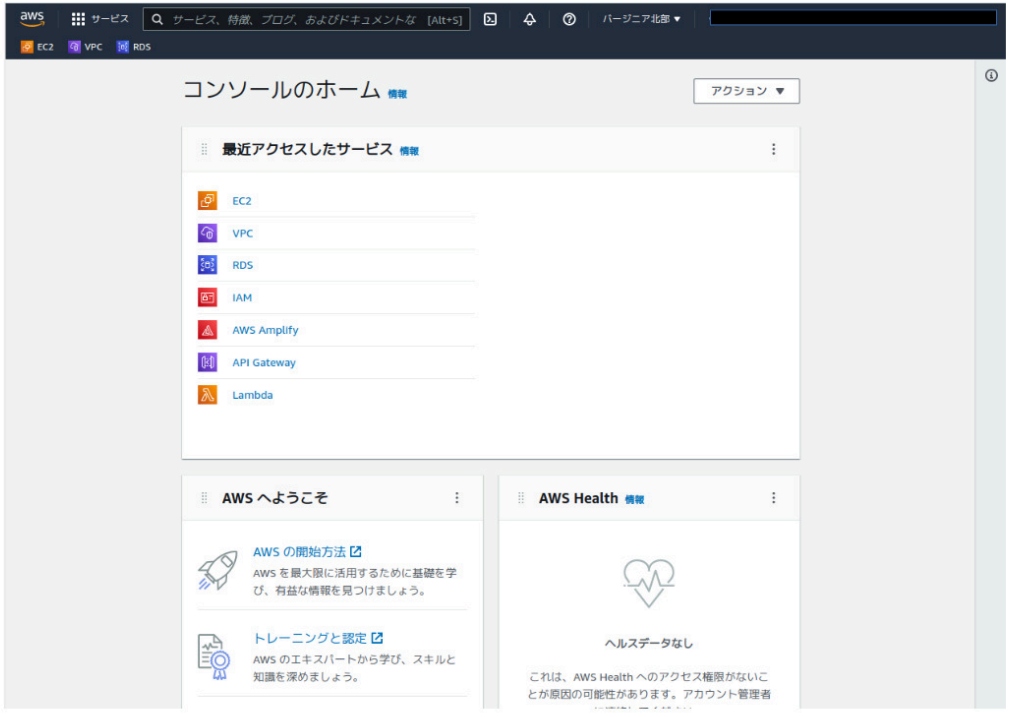
用語

- サーバ
  - サービスを提供するコンピュータのこと。「サーバクライアントモデル」も参照
- EC2 (発音：いーしーつー) ... Elastic Compute Cloudの略
  - AWSのサーバのこと。正確には仮想サーバ
- Public IP (発音：ぱぶりっくあいぴー)
  - サーバについている(インターネットから見える)IPアドレス; 午前中やった210.128.53.193の仲間
  - 反対語は(インターネットから見えてない)Private IPになりますが、 この話はネットワークの授業のだいぶ先で話します
- EIP (発音：いーあいぴー) ... Elasic IPの略
  - AWSで購入できるサーバに設定する(固定の)Public IPアドレス
  - [注意] EC2デフォルトのPublic IPは起動(vocareumでStart Labをクリック)するたびに異なります！

EC2の作成

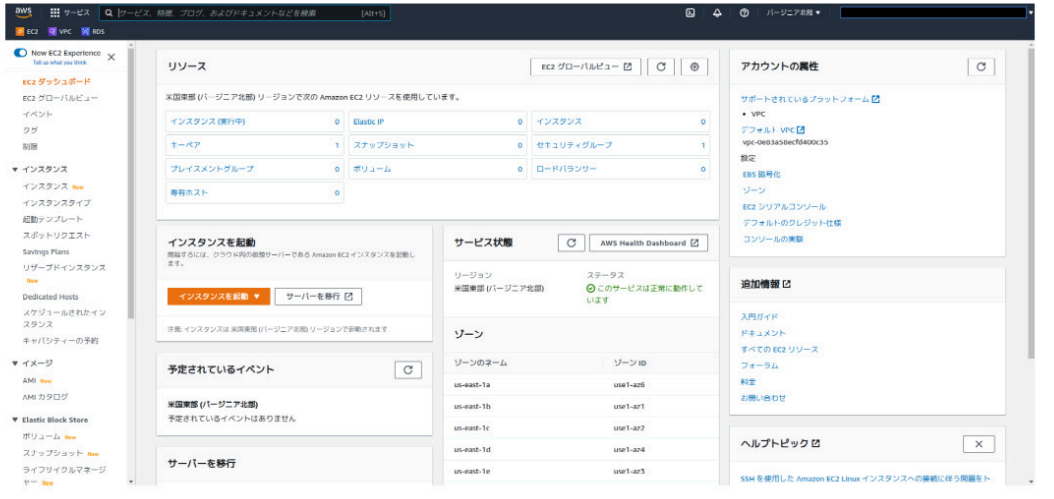
AWS Console -> EC2

1. AWS ConsoleでEC2をクリックして下さい。 EC2の文字が見つからない場合は検索してEC2サービスを探してください



EC2の管理画面でインスタンスを作成

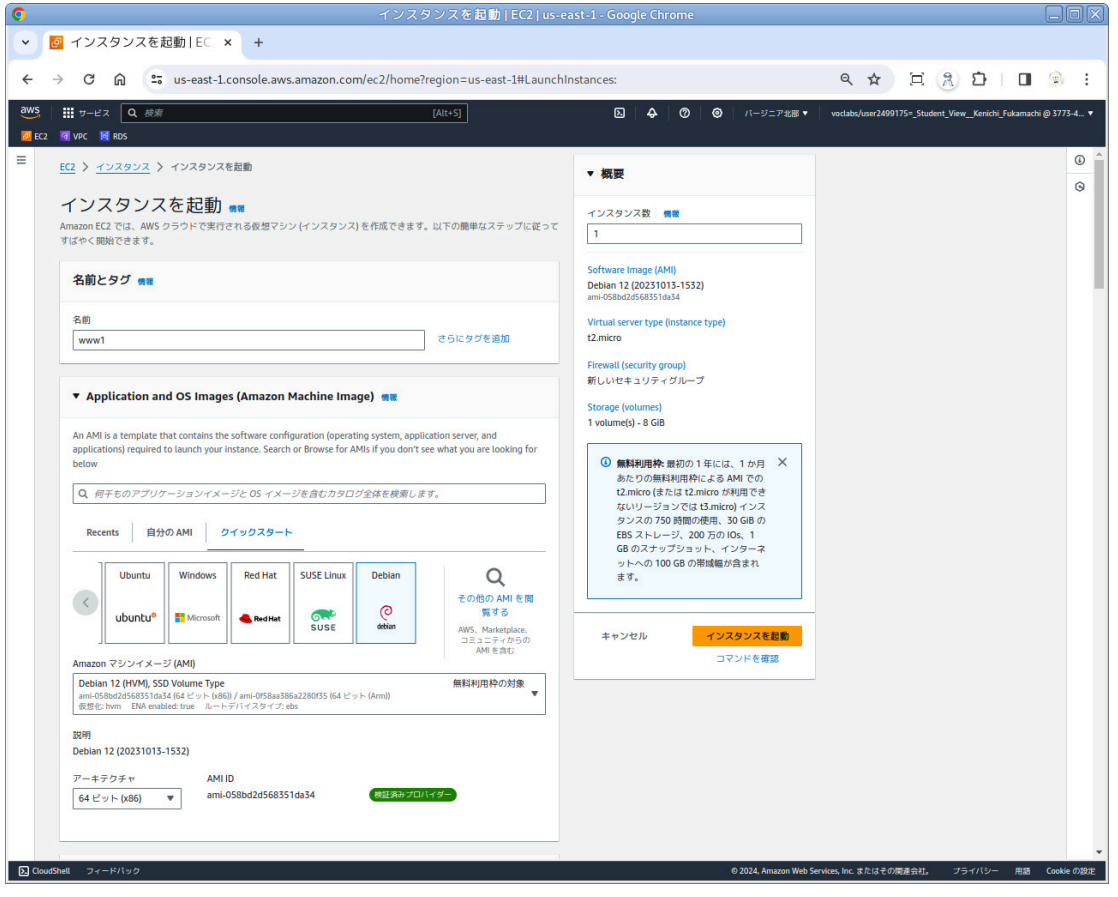
- オレンジ色の「インスタンスの作成」をクリックし、 Debian GNU/Linuxのサーバを一台作成します
- 仕様は以下のとおり
  - EC2につける名前は**学籍番号**
    (以下では例としてb2902900)
  - OSイメージはDebian GNU/Linux
  - EC2の種類はt2.micro
  - キーペアはvockeyを指定(脚注1)
  - セキュリティグループはSSH,HTTPS,HTTPの3つが使えるように設定(次回以降に利用する)
  - ストレージ(ディスク)はデフォルトでよい



(脚注1) SSHの鍵の指定です。 ここを忘れるとログインできません

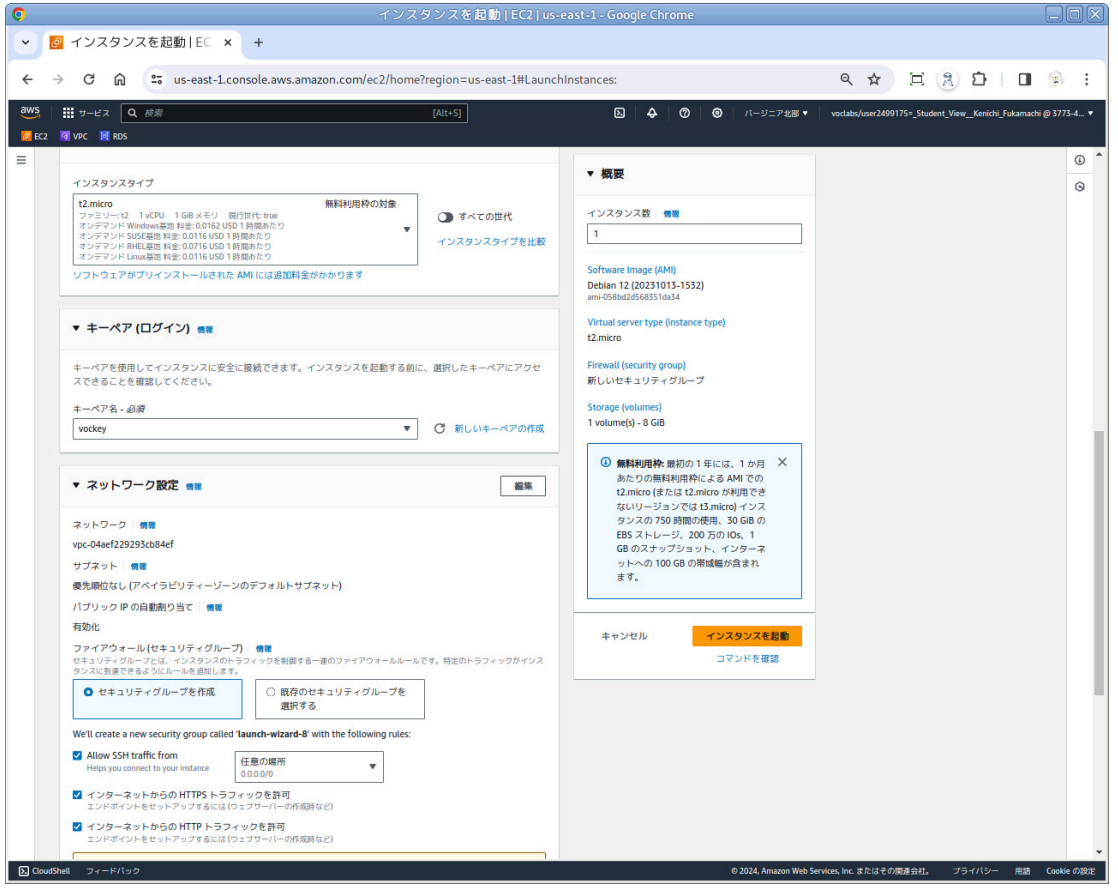
EC2の作成 詳細(1)

- EC2につける名前は**b2902900**
- OSイメージはDebian GNU/Linux
  - OS Imagesの右の方にあるので、右のほうを探してください
- EC2の種類はt2.micro



EC2の作成 詳細(2)

- **キーペアにvockeyを指定**(脚注1) 【重要】
- ファイアウォール(セキュリティグループ)は SSH,HTTPS,HTTPを許可してください
  - ようするに、すべてにチェックを入れてください



(脚注1) SSHの鍵の指定です。 ここを忘れるとログインできません

演習支援システム(Colitas)のインストール

Colitas = COmmand LIne Training Assitance System; CLI砂漠に咲く花が甘い香りを漂わせます,Hotel Calfornia で...

演習支援システム(Colitas)のインストール

EC2の作成後に行います

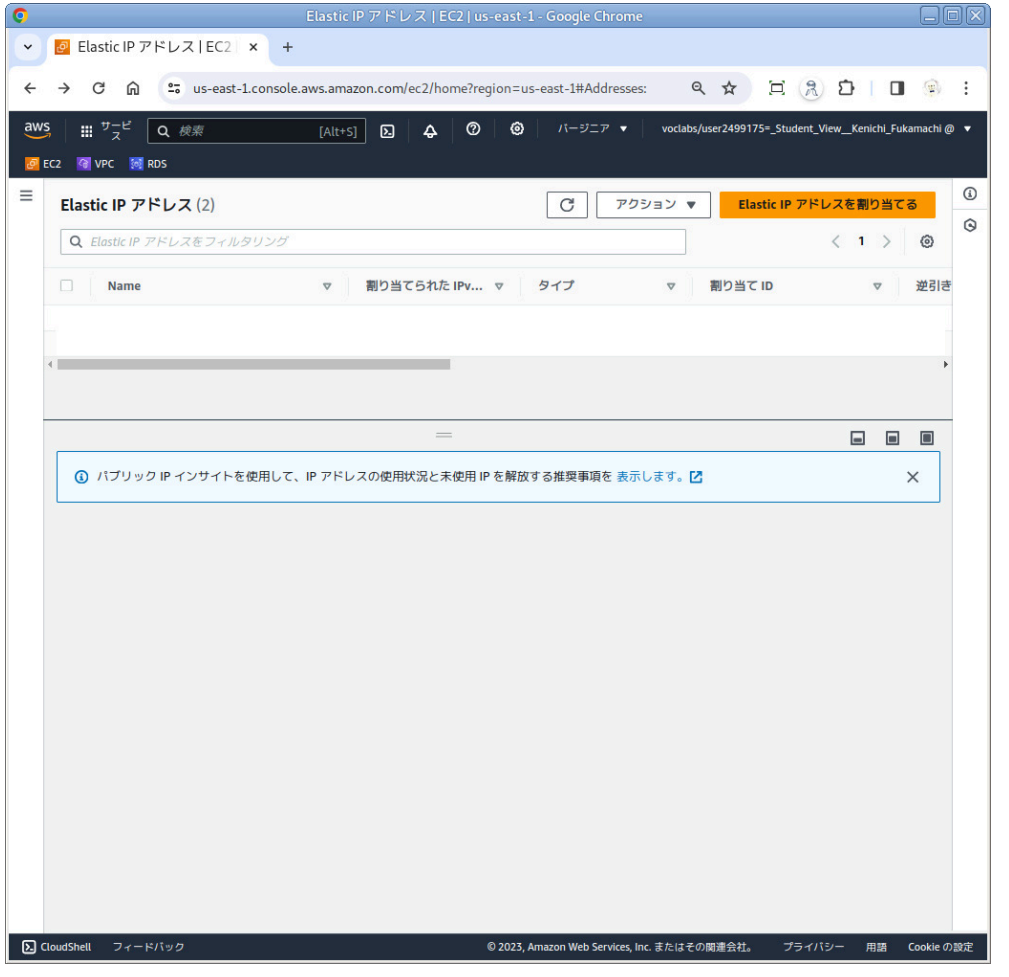
1. EC2の管理画面でPublic IPをコピー
2. https://karutoo.github.io/files2024/ にアクセス
   - 以下の情報を入力すると、謎の呪文が出力されるので、それをvocareumにコピー＆ペースト
     - 学籍番号
     - Public IPアドレス
     - パスワード(自分で好きなものを付けてよい)
   - 詳細はポータルにマニュアルを参照のこと
3. EC2でインストールが始まります。しばらく待ちます。EC2が再起動すると終了です
4. [課題] インストールされているか？を確認します
   - TA/SAさんに確認してもらってください

(脚注) システムの作者がデモしま〜す

Elastic IPの作成と関連付け

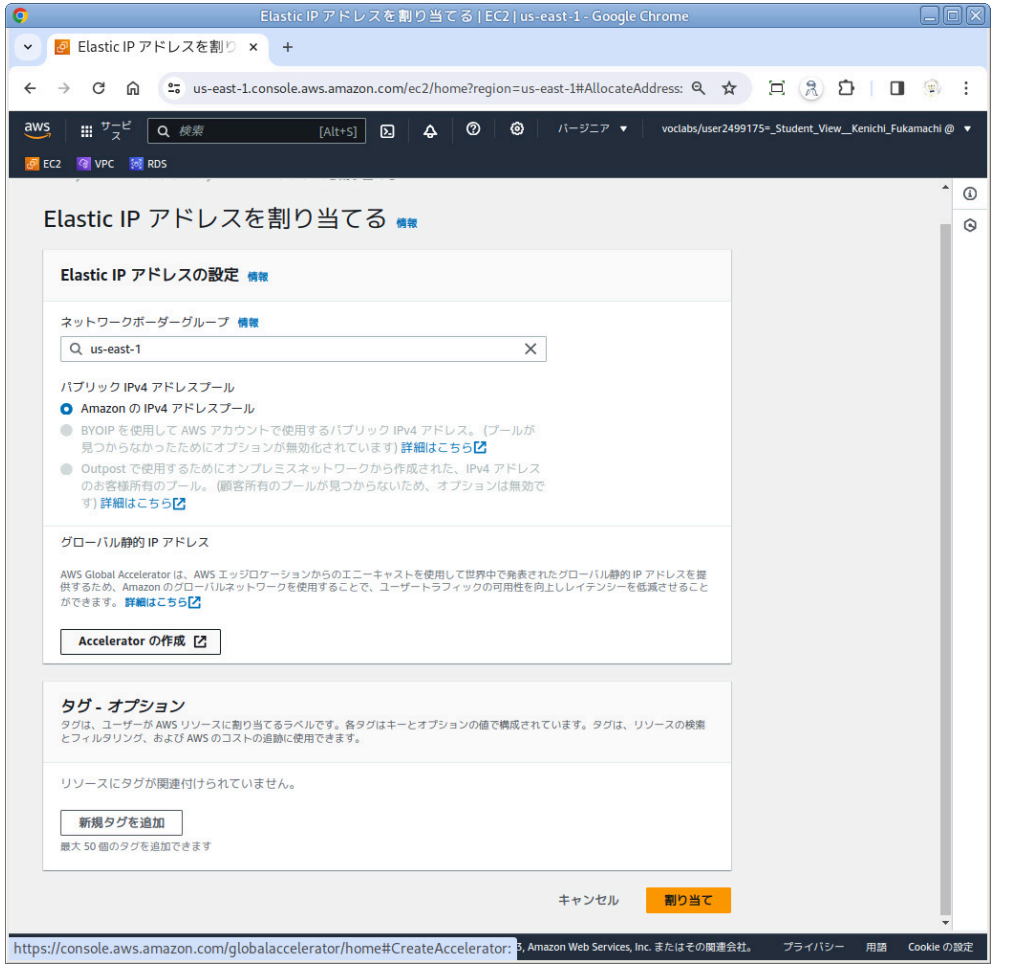
EIPを取得してEC2につける(1)

- Elastic IPを検索してクリック
- Elastic IPアドレスの管理画面で右上オレンジ色の「Elastic IPアドレスを割り当てる」をクリック



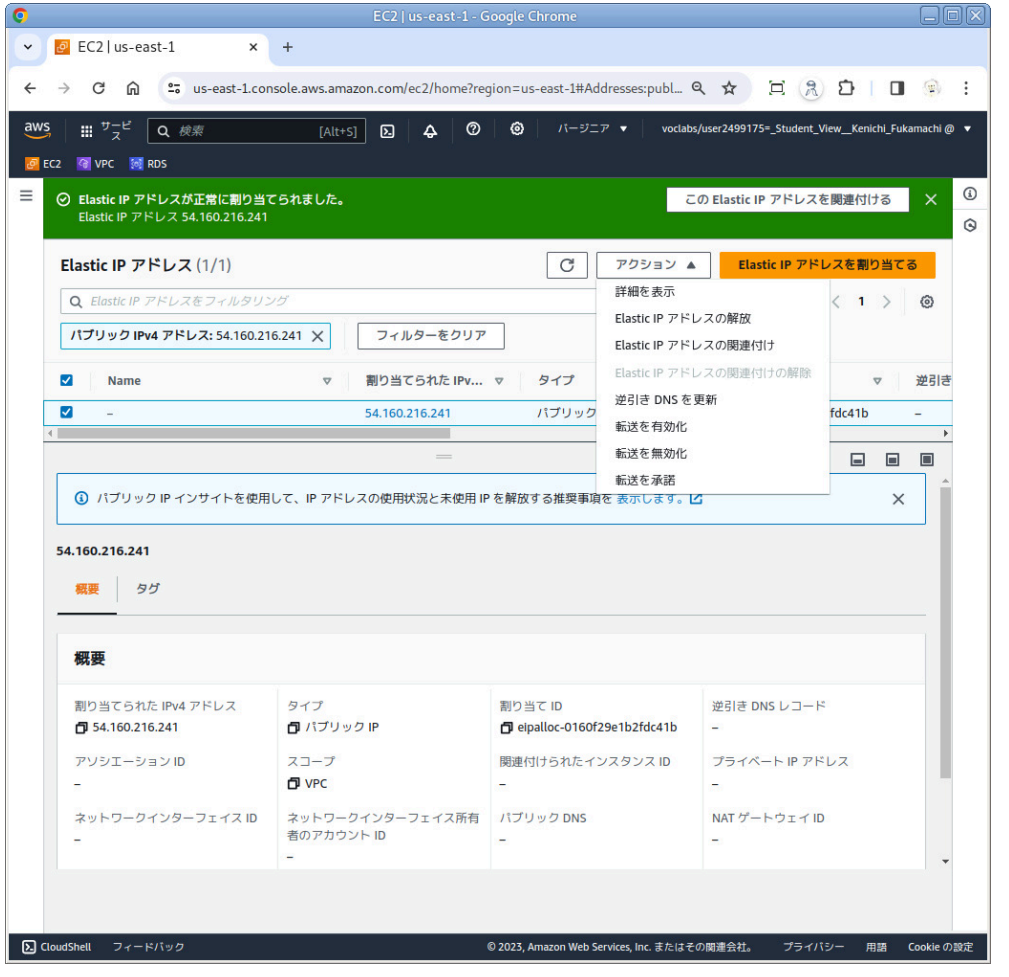
EIPを取得してEC2につける(2)

- 「Elastic IPアドレスを割り当てる」画面ではデフォルトのままでOK。 そのまま右下オレンジ色の「割り当て」をクリック



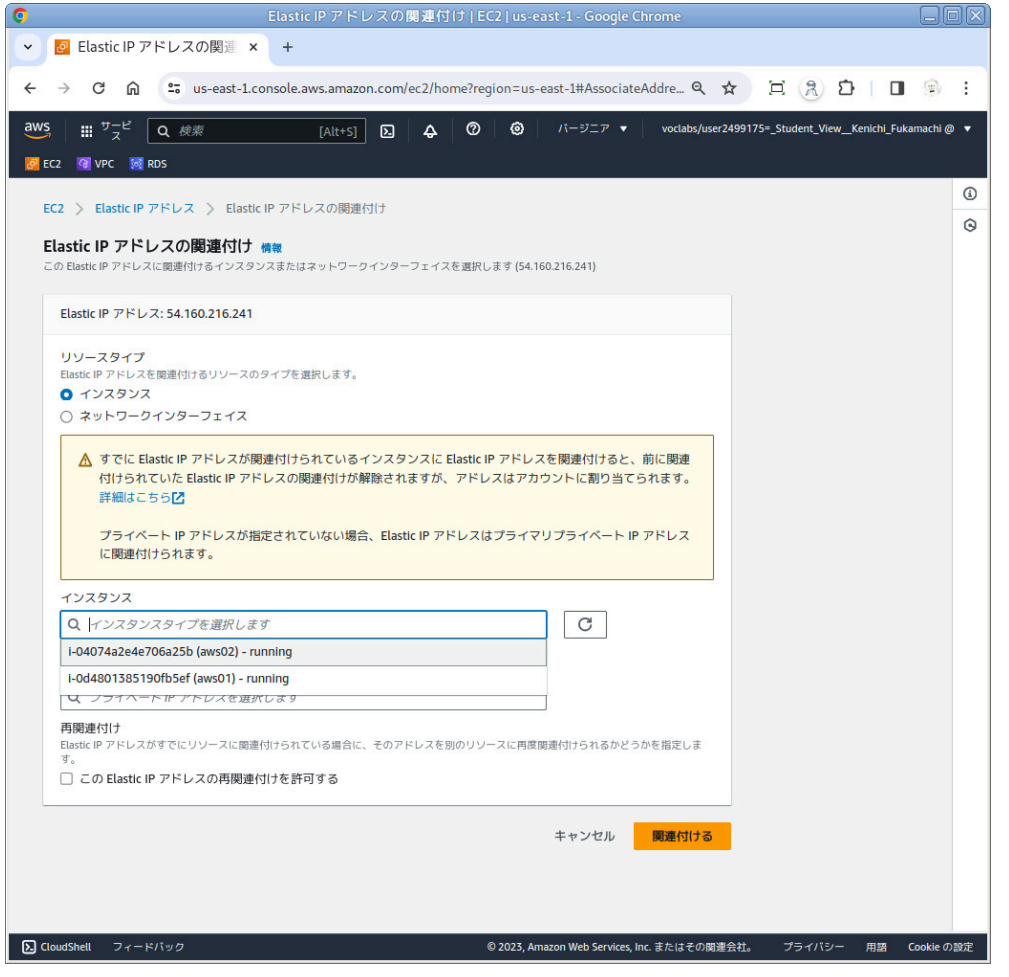
EIPを取得してEC2につける(3)

- Elastic IPアドレスの管理画面に戻るので、 いま割り当てたIPアドレスを選択し、「アクション」から「Elastip IPアドレスの関連付け」をクリック



EIPを取得してEC2につける(4)

- 「インスタンスを選択します」のあたりにマウスを持っていくと、
- EC2の候補が表示されるので
- 関連付けたいEC2を選択し
- 右下オレンジ色の「関連付ける」をクリック



情報システム開発基礎演習

ITインフラ編 第03回

3年、春学期、必修

本日の目標

1. 基礎的なUnix/Linuxコマンド操作ができる
2. WWWサーバをたてられる
3. 自分のホームページを作ることができる

お品書き

1. システム構築演習支援システム(Colitas)について
2. 午前中の復習
   - HTTPとWWWサーバのしくみ
3. 前回の復習
4. 用語(本日のNEW)
5. 演習 WWWサーバ(www.py)の作成
   - WWWサーバにはPythonで書いたwww.pyを配布します
6. [必須課題] (1)ホームページを編集する(2)構成図(ポータルのレポートボックスへ)
7. TA/SAさんに(1)を確認してもらったら終了(三々五々解散してOK)
8. [自由課題]「自分のホームページをリッチにする」

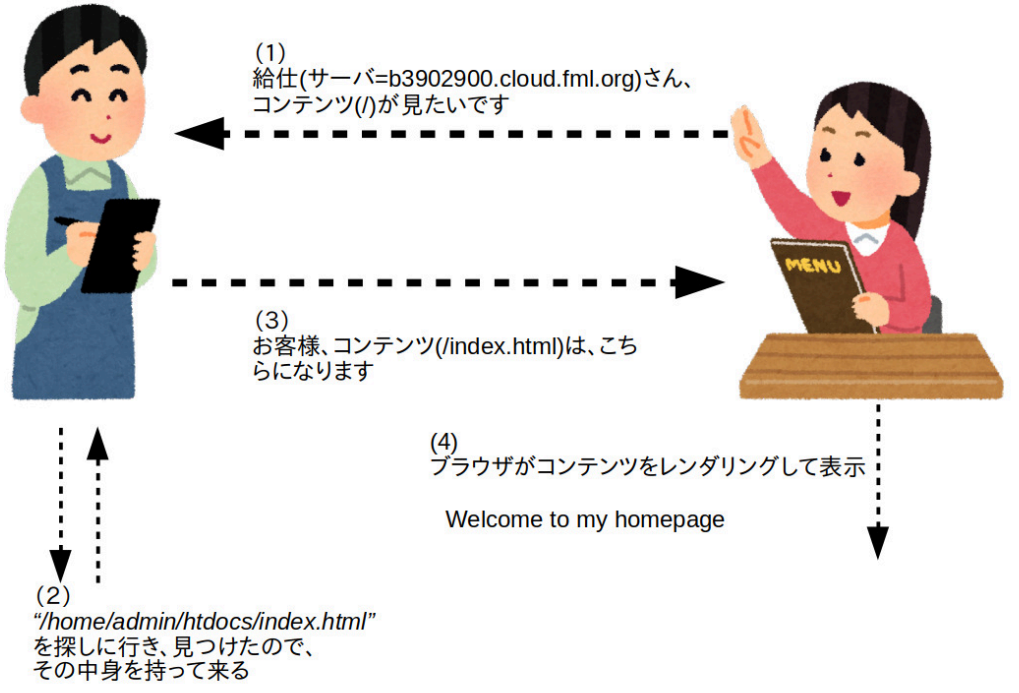
演習支援システム(Colitas)の解説

Colitas = COmmand LIne Training Assitance System; CLI砂漠に咲く花が甘い香りを漂わせます,Hotel Calfornia で...

復習

ホームページ（WWW）のしくみ

- (1) URL(`http://b2902900.cloud.fml.org/`)をブラウザでクリックすると
- (2) 【WWWサーバの内側の動作】URL右端の`/`を`/index.html`と解釈しindex.htmlの中身を返す準備をします
- (3) WWWサーバはブラウザにコンテンツ(index.html)を送り返します
- (4) ブラウザは受け取ったコンテンツを解釈して表示します(レンダリング)



(脚注) これを「サーバ(給仕)・クライアント(お客様)モデル」と呼んでいます

URLとは「サーバ名＋見たいコンテンツの場所」

URL = サーバ名＋見たいコンテンツの場所

例: `http://b2902900.cloud.fml.org/` = サーバ名(b2902900.cloud.fml.org) + 場所(/)

- (ここでは、午後の科目で使うWWWサーバの仕様で説明します)
- サーバ名が `b2902900.cloud.fml.org`
- サーバ名の右側部分(ここでは `/`)がサーバ上でのコンテンツの場所の指定
  (注: `/` = `/index.html` は相対位置です。 WWWサーバ(www.py)には `/index.html` の実在する位置は `/home/admin/htdocs/index.html` であると解釈する設定が入っています)

```
[例]
http://b2902900.cloud.fml.org/              ... デフォルトのコンテンツ /home/admin/htdocs/index.html
http://b2902900.cloud.fml.org/janken.html   ... この場合のコンテンツは /home/admin/htdocs/janken.html

http://portal.net.fml.org/
```

(脚注) www.pyとか、具体的なコンテンツ場所の説明は、月曜午後の科目にあわせています

用語

- プロトコル ... Protocol
  - 相互に取り決めた約束事。インターネットの場合は「通信規約」といった意味
- IPアドレス(発音：あいぴーあどれす) ... IP Address
  - コンピュータを識別する数字。 ちなみにIPはInternet Protocolの略
- コマンド
  - コンピュータに与える命令で擬似英語っぽい単語(+文章)、スペース区切りに注意
- AWS (発音：えーだぶりゅーえす) ... Amazon Web Services
  - もっとも有名な事実上の業界標準であるクラウドサービス
- Unixオペレーティングシステム (発音：ゆにっくす)
  - このテキストではDebian GNU/Linuxのことです
- Linux　（発音：りなっくす、りぬ〜くす）
  - クラウドサービスでもっともよく使われるOS
  - UNIXのニセモノ(Unix クローン)のひとつ

(脚注) 先週分および午前中の授業と一部重複しています。午前の復習も兼ねているわけですね

用語

- サーバ
  - サービスを提供するコンピュータのこと。「サーバクライアントモデル」も参照
- EC2 (発音：いーしーつー) ... Elastic Compute Cloudの略
  - AWSのサーバのこと。正確には仮想サーバ
- Public IP (発音：ぱぶりっくあいぴー)
  - サーバについている(インターネットから見える)IPアドレス; 午前中やった210.128.53.193の仲間
  - 反対語は(インターネットから見えてない)Private IPになりますが、 この話はネットワークの授業のだいぶ先で話します
- EIP (発音：いーあいぴー) ... Elasic IPの略
  - AWSで購入できるサーバに設定する(固定の)Public IPアドレス
  - [注意] EC2デフォルトのPublic IPは起動(vocareumでStart Labをクリック)するたびに異なります！

[解説] 前回のColitasのインストール、Colitasの使い方

- TA/SAさん、よろしく

演習

Colitasのメッセージを参考にがんばってみよう

はじめるまえに

- プレテスト

演習「自分のホームページを立てる」

- １５～２０分の時間をあげます。以下の演習を行ってください
  - 演習「自分のホームページを作成してください」
  - 課題「自分の学籍番号が表示されるホームページにしてください」
- ふだんスマートフォンでもアプリを入れたりしますよね？そのへんを思い出しながら、**作業手順を自分で考えてください**
  - 作業のヒントや必要な要素は続くスライド(合計２ページ)に書いてあります
  - また実際にコマンド操作をする際にはColitasの支援を受けられます
  - WWWサーバはwww.pyというPythonスクリプトを配布しています。それを使ってください

Colitasのメッセージを参考にがんばってみよう

演習「自分のホームページを立てる」

- 自分のホームページを確認する方法は「ブラウザに URL を入力してENTER」です
  - ホームページのURLは `http://学籍番号.cloud.fml.org/` です
- ダウンロードするコマンドは `curl -O URL` です
  - URL は http://api.fml.org/dist/www.py
- www.pyを実行するコマンドは`sudo python3 ファイル名`です
  - ここでファイル名は www.py のことです

Colitasのメッセージを参考にがんばってみよう

必須課題「自分のホームページにする」

- 自分のホームページの表示を「Welcome to 学籍番号」に変更してください
  - 例: 学籍番号 b2902900 の人は「Welcome to b2902900」
  - 編集するコマンドは`nano ファイル名`です
  - このファイル名は /home/admin/htdocs/index.html です

Colitasのメッセージを参考にがんばってみよう

終わったら？

- (早々とおわってしまったら、同期をとるので、すこし待っててください)
- ポストテスト
- 口頭試問

ラウンド２

はじめに

- スマートフォンのたとえに戻りますけどね
- スマートフォンのアプリを入れるときって、どういう手順でしょうか？

1. アプリを〜する
2. アプリを〜する
3. アプリを〜する

情報システム開発基礎演習

ITインフラ編 第04回

3年、春学期、必修

本日の目標

1. HTMLの基礎を理解する
2. WWWサーバと会話できる
3. (会話の応用として)ジャンケンができる

(脚注) WWWサーバとの会話とは、WWWサーバと「データのやりとりが出来る」という意味です

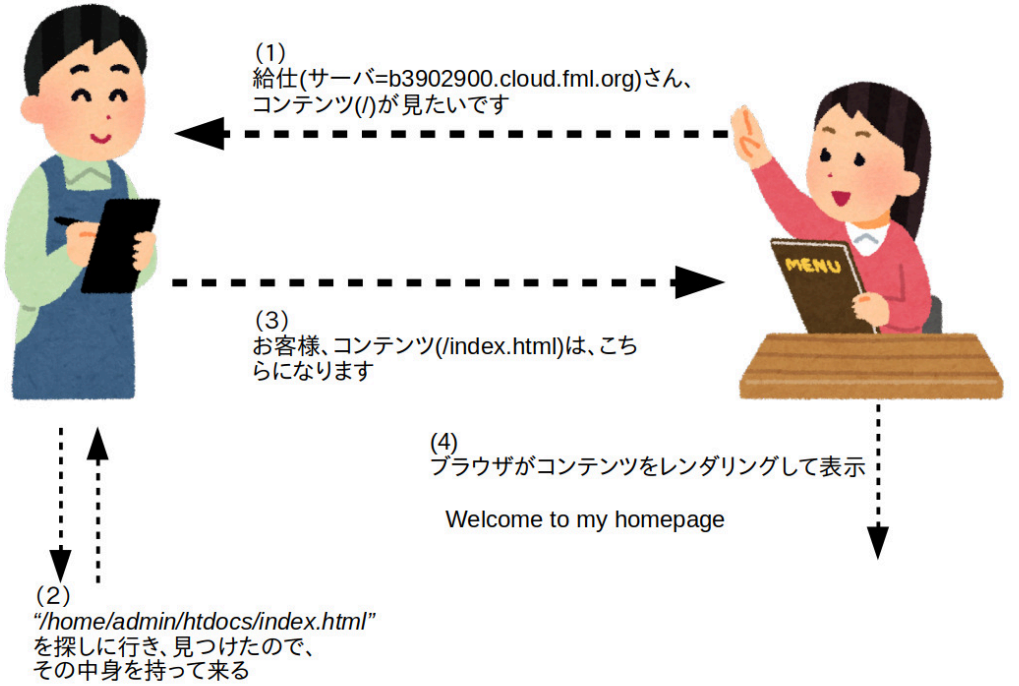
おしながき (#04)

1. 解説: HTML (1)
2. 演習「WWWブラウザでジャンケンするHTML」を作る
3. [必須課題] ジャンケンのフローチャート、janken.html
4. TA/SAさんにフローチャートとジャンケンの実演を確認してもらい終了(三々五々解散してOK)
5. [自由課題]「ISBN検索」(出来る人は一瞬で出来るはずなので、時間中に、みせてください)

解説: HTML(1)

復習: ホームページのしくみ

- (1)
- (2)
- (3)
- (4)



前回のテキストより … あえて空白にしてあります。分かるかな？って、図に説明が書いてありますけどね;-)

HTMLの基本(1): タグ

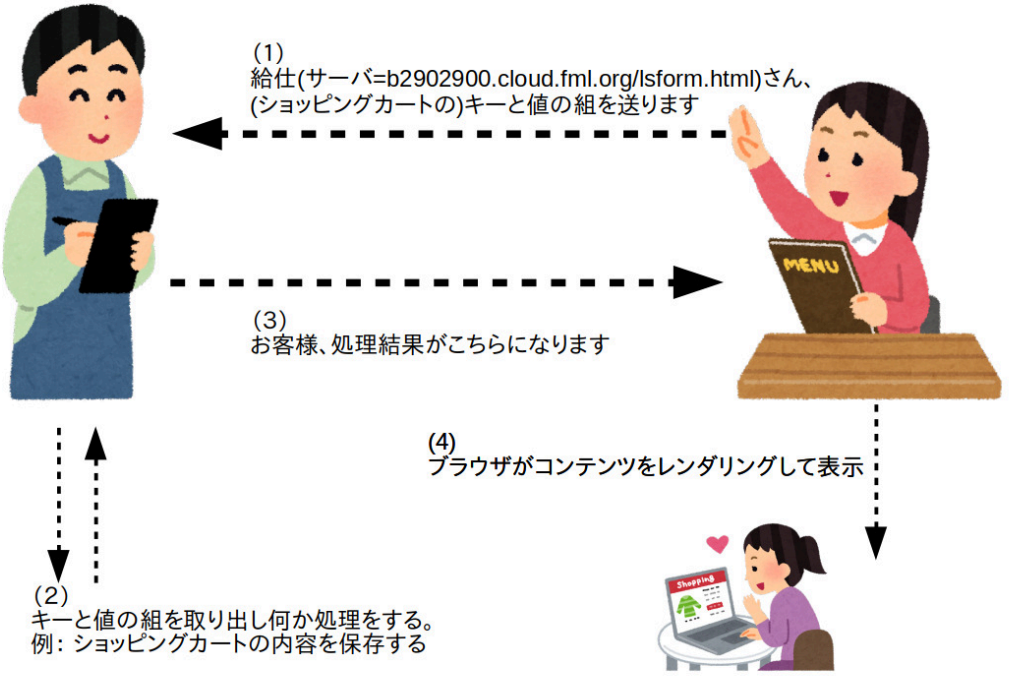
```
<P> 新しい段落(paragraph)を始めます。ブラウザは、改行など適切な見栄えになるようレンダリングを行います

<P> <B>このようにBタグではさむと、太い文字(ボールド体)</B>で表示されます
```

- 単なるテキストです。ただしタグ(`<tag>`)と呼ばれる命令を含めることが出来ます
- タグの始まり`<tag>`と終わり`</tag>`を書くと、それらのタグで挟まれた間だけで命令が有効になります
  - **閉じる方のタグは`</`(小なり+スラッシュ)**で始めます
- 動作確認だけ出来ればよい授業なので、**とりあえず`<P>`だけは覚えましょう。読みやすくなるから**

(脚注1) 真面目にHTML5をやると大変すぎるので初期のHTMLをやります (脚注2) この例で分かるように、 HTMLのタグには文体の「構造」(P)と「装飾」(B)という異なる概念が一緒くたにされています。 これが批判されて、厳密化され、現在のHTML5という読みづらい謎言語になりました。 謎すぎるので授業ではやりません。ウエブデザイナーになりたい人は勉強してください

HTMLの基本(2a): FORM文(CGI)

- たとえばショッピングカートを作ることを考えます。 カートがあつかう情報とは「商品名」と「数量」の組です。組は複数可
- 元祖HTMLでは、こういうショッピングが出来ないため、新たに **「キーと値の組をサーバに送る」約束事**を作りました。 **この取り決めがCGI** (CGI = Common Gateway Interface)
- WWWブラウザとサーバ両方がCGIをサポートしている必要があります (過去30年以上そうなっているので安心して使ってOK)

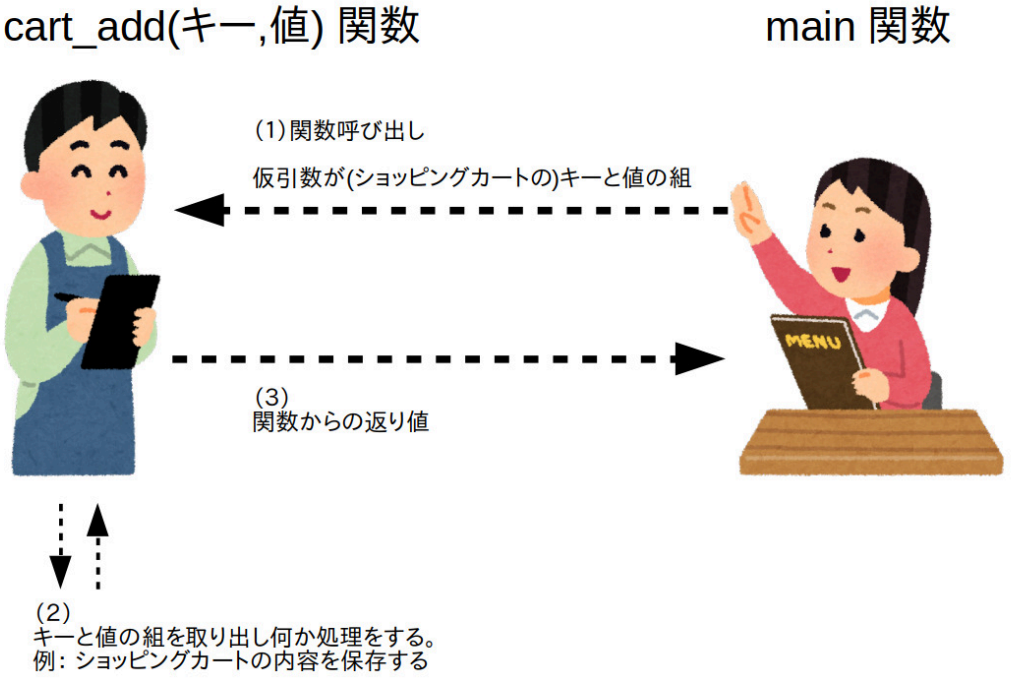


(脚注) ショッピングカートの最初のところだけです

復習: C言語の関数呼び出しを思い出そう

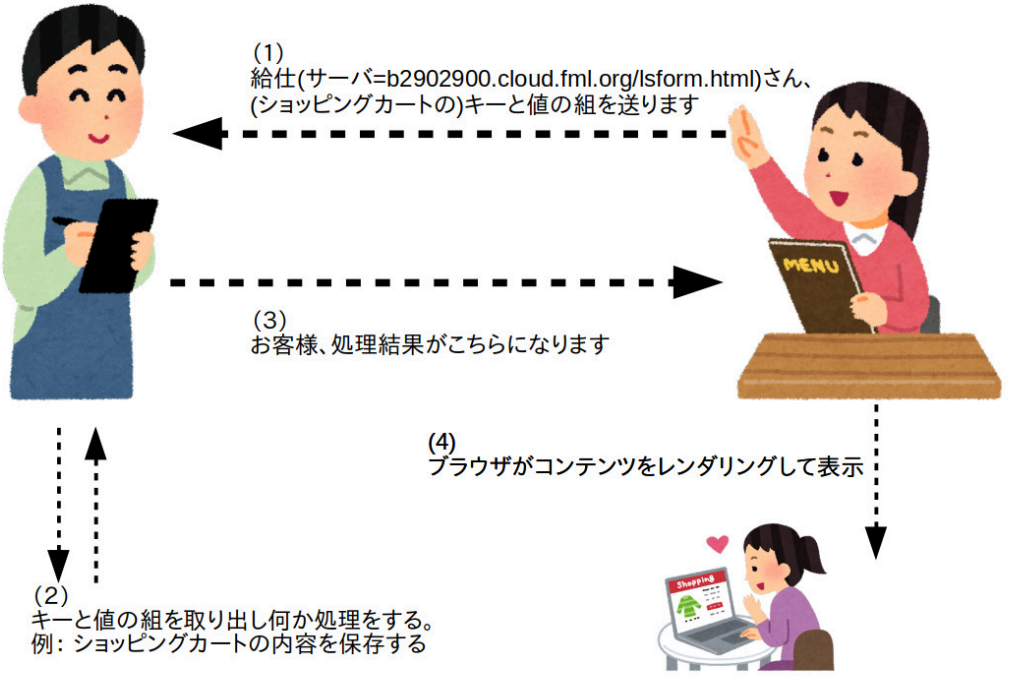
```
(ソースコードのイメージ)

// カートに商品と数量を入れる
cart_add(item1, 1);
cart_add(item2, 2);
cart_add(item3, 3);

// 画面にショッピングカートの中身を表示する
cart_show();
```

- **処理を依頼するURLが関数に相当**します
- だいぶ文法は異なりますが、 C言語の関数呼び出しと同じですよね？
- CGIの場合、たいてい関数が別のPCで実行されるけど、**図としては同じですよね？**



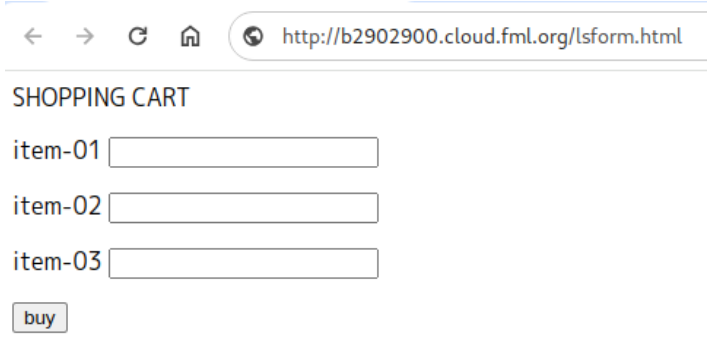
HTMLの基本(2b): FORM文(CGI)

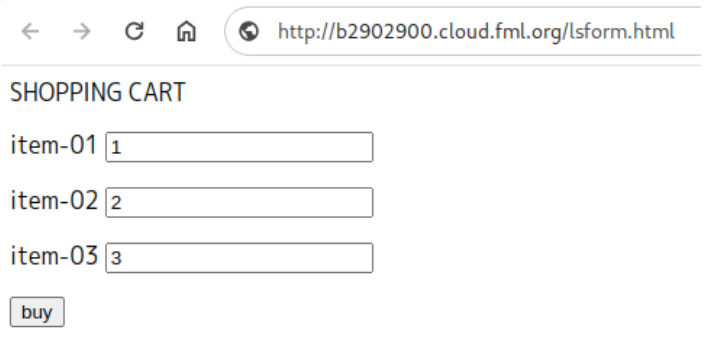
- FORM文で、キーと値(「商品名」と「数量」)の組をサーバに送るHTMLを書けます
  - ブラウザはFORM文に沿って入力欄や送信ボタンなどを作成(レンダリング)します
  - ブラウザはキーと値の組をブラウザに送る
  - サーバ側では、キーと値の組を取り出す
- 上記までがCGIの仕様で、 たいてい取り出した後、**なんらかの処理をして結果を返すサーバの作りこみが、別途、必要**です
  (最近は、このサーバのことをWeb APIと呼ぶ)

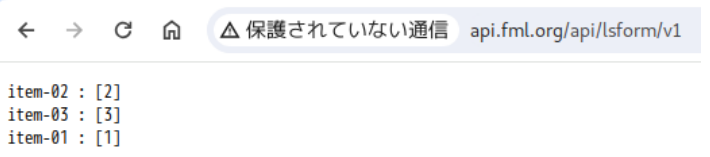


(脚注) ショッピングカートの最初のところだけです

FORM文の体験例: http://b2902900.cloud.fml.org/lsform.html







(脚注1) このあと演習でlsform.htmlをダウンロードして実際に体験してもらいます (脚注2) 図(右)の画面はサーバごとに異なります。 これはlsform.htmlのaction=URLで指定されているURL「http://api.fml.org〜」で動いているサーバが、 ブラウザから送られてきたキーと値の組を返している様子です。 ちなみに、**演習のデモ等で使うサーバは、すべてGo言語で書かれています。 これは「キーと値の組」を取り出して%v:%vでGo言語がよしなにフォーマットした表示**になっています (注: 内部では値をsliceで保持しているため[数字]と表示)

HTMLの基本(3a): lsform.htmlの解説 (画面の表示)

「画面に表示される文字」と「入力欄や送信ボタンの作り方」という２種類の話が混ざっていることに注意

HTMLの基本(3b): lsform.htmlの解説 (サーバの指定)

```
<form method="POST" action="http://api.fml.org/api/lsform/v1">
   〜省略〜
</form>
```

- FORMタグ(`<form>`)で、データのやりとりの詳細設定を書きます
  - この部分はブラウザでレンダリングされませんが、
  - WWWサーバとのやりとりで重要な情報を書くところです
- タグのなかにはタグの属性等が書けます。スペース区切りで`属性=値`形式です
  - この例でも`<FORM 属性1=値1 属性2=値2 ...>`となっています
  - `method=POST`はHTTPの動作モードの指定と考えてください
    FORM文では、たいていPOSTを指定するので、**授業ではmethod=POSTと覚えておいてOK**
  - `action=URL`が使うサーバのURLの指定です。ここは様々です。C言語で呼び出す関数を変えれば色々できるように、 **URLを変えればイロイロな処理が可能となります**

HTMLの基本(3c): lsform.html (そのもの) 【オマケ】

```
<P>SHOPPING CART
<form method="POST" action="http://api.fml.org/api/lsform/v1">
  <P>item-01
    <input name="item-01"  type="text">
  <P>item-02
    <input name="item-02"  type="text">
  <P>item-03
    <input name="item-03"  type="text">
  <P>
  <input type="submit" value="buy">
</form>
```

- これが、このあとダウンロードするlsform.htmlです

(脚注) このあと実際に使うlsform.htmlの解説です(1)

HTMLの基本(3d): lsform.htmlの読み方 【オマケ】

```
<P>SHOPPING CART                                    <!-- ここは普通に表示されます(改行あり)              -->
<form method="POST" action="Web APIサーバのURL">    <!-- methodはHTTPの動作モードのようなもの,大抵POST -->
  <P>キーの名称1                                    <!-- ここは普通に表示されます(改行あり)              -->
    <input name="キーの名称その1"  type="text">      <!-- 入力欄としてレンダリングされます                -->
  <P>キーの名称2                                    <!-- ここは普通に表示されます(改行あり)              -->
    <input name="キーの名称その2"  type="text">      <!-- 入力欄としてレンダリングされます                -->
  <P>キーの名称3                                    <!-- ここは普通に表示されます(改行あり)              -->
    <input name="キーの名称その3"  type="text">      <!-- 入力欄としてレンダリングされます                -->
  <P>                                               <!-- 改行するために無駄にPを入れました               -->
  <input type="submit" value="送信ボタンの表示名">  <!-- 送信ボタンとしてレンダリングされます             -->
</form>
```

- `<!- コメント ->`で説明しています。 `<P>～`の部分がないと箱が並ぶだけでページの意味が分かりません。「説明を書く元祖HTML」と「FORM文が作る入力欄」の両方を混ぜこぜで書いていきます
- 「キーの名称1」を「変数名1」と考えると、関数呼び出しのイメージと重なります

(脚注) このあと実際に使うlsform.htmlの解説です(2)

演習

演習内容 (概要)

1. FORM文のテンプレート `lsform.html` をダウンロードし、FORM文を体験してください
2. Web API版ジャンケンのフローチャートを書いてください
3. lsform.htmlをコピーして、ファイル名janken.htmlを作成します
4. janken.htmlファイルを編集し「Web API版ジャンケン」を完成してください

- もう少し詳しい説明が次ページ以降にあります

(脚注) ステップの1,3,4は Colitas が支援してくれます

演習内容(1) lsform.htmlをダウンロードし、FORM文を体験する

- FORM文のテンプレート `lsform.html` をダウンロードしてください
- ダウンロードするコマンドは `curl -o ファイル名 URL` です
  - オプション `-o` は、マイナス、アルファベット小文字のo(oscarのo)です
  - (ダウンロード先の)ファイル名 = /home/admin/htdocs/lsform.html
  - (ダウンロード元の)URL = http://api.fml.org/dist/lsform.html
- ブラウザで各自のWWWサーバ `http://学籍番号.cloud.fml.org/lsform.html` にアクセスし、「FORM文でキーと値を送る」体験をしてください。
  - 体験イメージはスライド p.10 を参照

(脚注) Colitas が支援してくれます

演習内容(2) Web API版ジャンケンのフローチャートを書きなさい

- フローチャートは1年前に「プログラミングとアルゴリズム基礎」で習いました
  - ELの教科書「大学情報 > 情報工学(mobile対応) > アルゴリズム > フローチャート」を参照
- STARTは「lsform.htmlをブラウザが読みこんだところ(レンダリング前)」とします
- 「WWWサーバ呼び出し」を「関数呼び出し」(つまり「定義済み処理内容」)と考えてください

演習内容(3) lsform.htmlをコピーしてjanken.htmlを作成する

- ファイル/home/admin/htdocs/lsform.htmlをコピーしてください
- コピー先のファイル名は/home/admin/htdocs/janken.htmlです
- Unixターミナルでファイルをコピーするコマンドは cp (CoPyの頭文字)です

```
[cpコマンドの書式]

cp コピー元 コピー先
```

(脚注) Colitas が支援してくれます

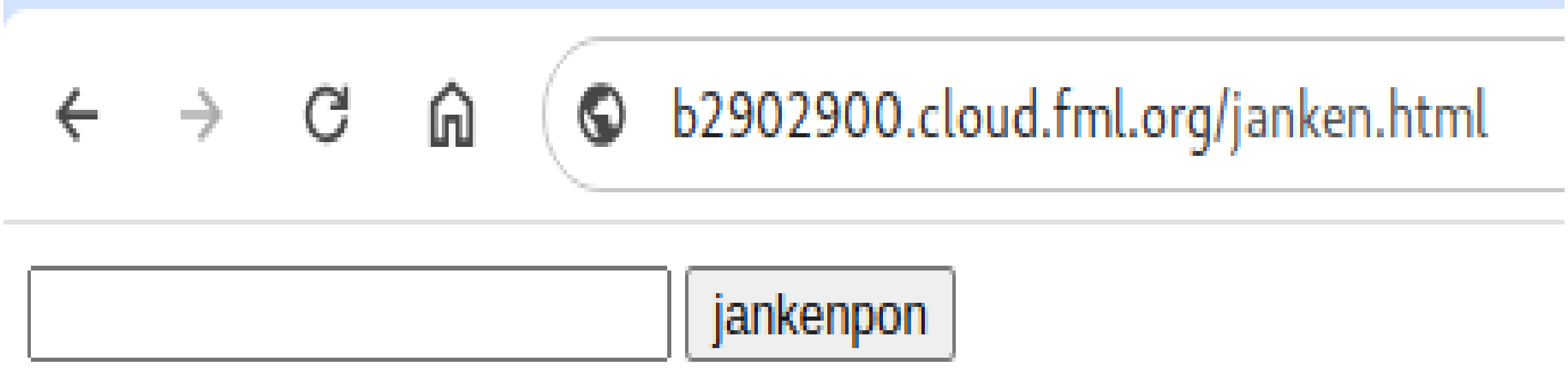
演習内容(4) janken.htmlファイルを編集する

- janken.htmlファイルを編集し「ジャンケンサーバ(Web API)を利用したジャンケン」を実現してください
- 作成する画面イメージは次ページを参照
- ジャンケンサーバの仕様
  - グー、チョキ、パーを0、1、2で表現しています(C言語の演習でやりましたよね?)
  - URL = http://api.fml.org/api/janken/v1
  - サーバは、キーが `jibun`、その値として数字(0, 1, 2)が送られてくると期待しています
    (jibunは「ジャンケンの自分の手」という意味です)
  - サーバが返す「aiteはコンピュータ(相手)の手、kekkaは勝敗」です

    ```
    kekka = (3 + jibun - aite) % 3
    ```

(脚注) Colitas が支援してくれます

見本: Web API版ジャンケンの画面



見本: Web API版ジャンケン実行後のブラウザ画面

{"jibun":1,"aite":0,"kekka":1}

(脚注) これはJSON形式です。 JSONになっているのは、これを利用した自由課題のためです(今回の自由課題ではありませんが)

必須課題

次の二つをTA/SAさんに確認してもらってください

1. Web API版ジャンケンのフローチャート
2. 作成した janken.html が実際に動くところ
    - とうぜんURLは各自ことなります
    - URL = `http://学籍番号.cloud.fml.org/janken.html`

自由課題 (任意課題,講義時間内のみ受付 - 16:30)

- ISBN検索ができるページを作成してください
  - URLは `http://学籍番号.cloud.fml.org/isbn.html`
  - 使えるサービスはopenBD (open book database)の https://openbd.jp/
  - 使い方はopenBDのページを読んで勉強してください。 なお、openBDのページでは説明されていませんが、POSTメソッドも対応しています
  - 確認用の ISBN = 9784621066089
- 出来るひとは一瞬で出来ると思いますので、演習の時間内部でTA/SAさんに見せてください

情報システム開発基礎演習

ITインフラ編 第06回

3年、春学期、必修

おしながき (#06)

第6回は口頭試問をします

1. 第5回の宿題は、第5回のレポートボックスで提出してください
    - 忘れている人、いま出してください(-13:30)
2. [必須] Rekognitionのフローチャート
    - [自由,任意,加点対象]は別途みます
3. 答え合わせ (15:30- 予定)
    - かるく課題をやってもらいます
    - 回収
    - 解説編
4. ITインフラ編おわり
    - もっときちんと構築するの科目が秋学期の「クラウドコンピューティング」
    - 来週は休講、6/3から機械学習編へバトンタッチです