



GEEK JOB

キャンプ

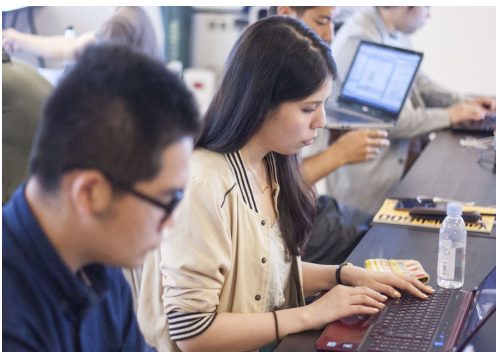
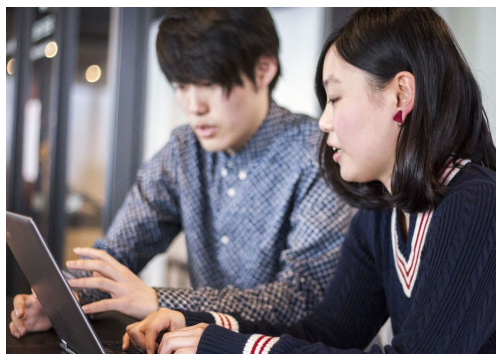
本日はお忙しい中、GEEK JOBキャンプへお越しいただきまして
誠にありがとうございます。

- ・プログラミングに触れる、楽しさを実感する
- ・ GEEK JOBキャンプの雰囲気やまなび方を体感する

こちらの2つを楽しみながら体験して頂ければと思います。

短い時間ではありますが、
どうぞ宜しくお願い致します。

GEEK JOBキャンプ運営



【1日体験タイムスケジュール】

10:00 ～ 11:00	体験前アンケートの記入、朝会の見学、パソコンの設定、午前中の課題実施
11:00 ～ 12:00	プログラミングキャンプオリエンテーション
12:00 ～ 13:00	お昼休憩
13:00 ～ 15:00	午後の課題実施
15:00～15:30	プログラミングキャンプのカリキュラム、教材紹介
15:30 ～ 18:00	引き続き課題実施（※希望者のみ） ★希望される方には個別面談も実施させていただきますので、お気軽にお声がけください。
17 : 00～	簡単な課題解説（※希望者のみ） 体験後アンケートの記入

【重要】アンケートのご協力をお願い致します。

体験前

<https://goo.gl/SCPwJX>

体験後

<https://goo.gl/rDvsu5>

＜午前中の課題＞

簡単なチュートリアル形式の課題を解いて頂き、実際に書いたコードを動かすことでプログラミングを体感して頂きます。書いてある順番通りに取り組んで頂くタイプの課題となります。

＜午後の課題＞

「世界のナベアツ」のネタをもとに、繰り返し処理や条件分岐を用いてソースコードを一から書く課題に取り組んで頂きます。

＜応用課題＞

応用課題の内容は、こちらからある程度出来上がったソースファイルを渡し、その内容をコーディングで直接修正するといった課題になります。ブラウザ上で見れる電卓を用意していますので、その「小数点ボタン」や「税抜きボタン」などを実装して頂きます。資料にはコードの書き方は載っておらず、修正対象のソースコードから意味のありそうな文脈を見抜き、それを模倣し、活用していくタイプの課題です。（ソースコードは講師から渡しますのでお声がけください）

＜GEEK JOBキャンプの学び方について＞

一般的なスクールとは異なり、「反転学習」「自発的/自律した学習」を教育理念としています。

「反転学習」・・・自宅で予習/復習→キャンプで実践/質問をして解決する
「自発的/自律した学習」・・・従来の受け身での学び方ではなく、
自ら質問を取りに行き課題を解決する学習の仕方

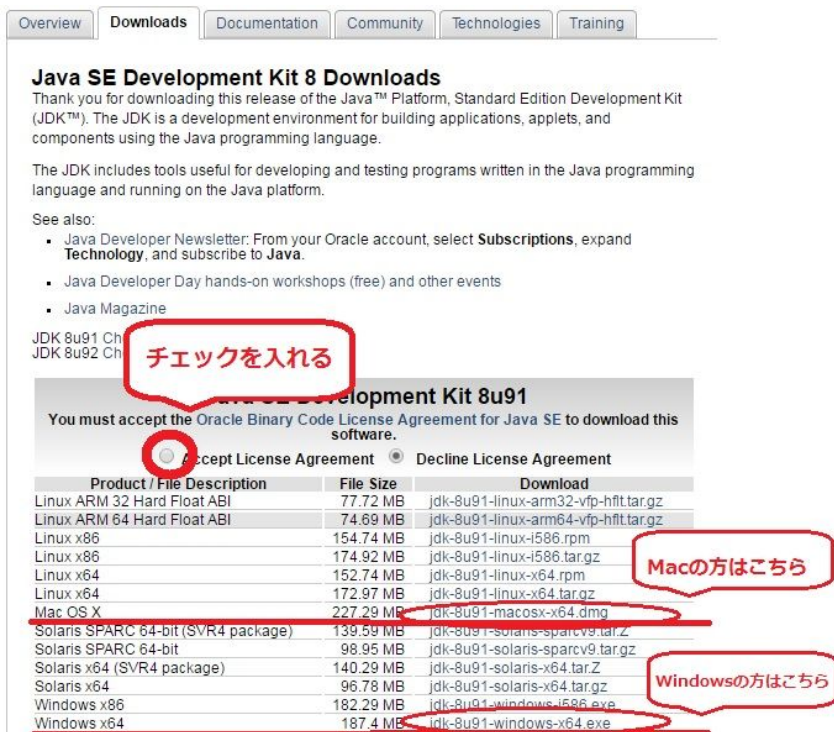
本日の一日体験ではGEEK JOBキャンプの学ぶ姿勢も体感して頂ければと思いますので、遠慮無くメンターや他のキャンプ生に積極的に質問してみてください。キャンプ生も日頃から互いに教え合っているの、質問されることにはとても慣れていて、むしろ歓迎している人たちばかりです。

【環境設定】

<Windowsの方>

①<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

にアクセスして、右の図の通りにチェックを入れてからインストールをする。その後は特に何もいじらず、「次へ」を押して進む。



②<https://www.apachefriends.org/jp/index.html>

からXAMPPをダウンロードし、インストールしてください。

XAMPPがインストールされたフォルダ内の「tomcat」フォルダを開いてください。

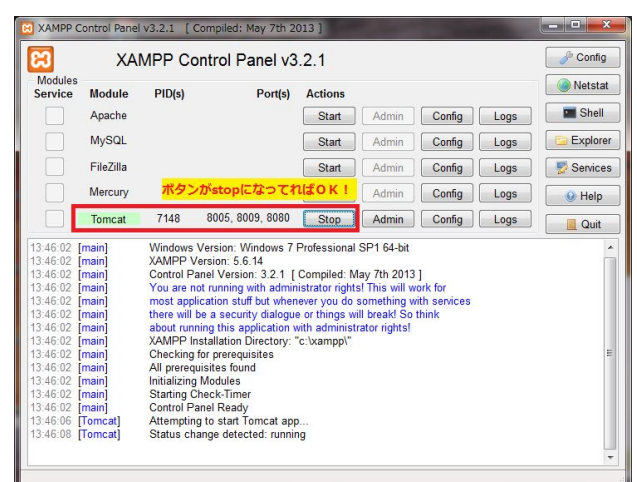
その中の「webapps」フォルダの中に、「camp」というフォルダを作成してください。

今回は、基本的にこの「camp」フォルダに自分のプログラムを置きましょう。

ここに配置したjspファイルは、XAMPPのコントロールパネルでtomcatを起動した後、下記の様に呼び出せます。

例)

campフォルダにtest.jspを置いた場合、



ブラウザを開いてURLバーに以下のようにアドレスを書いてください。
http://localhost:8080/camp/test.jsp

これで動作確認が行えるはずです。
同じ要領でjspファイルを作成し、どんどんコーディングしてみましょう！

<Macの方>

①<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

にアクセスして、右の図の通りにチェックを入れてからインストールをする。その後は特に何もいじらず、「次へ」を押して進む。

The screenshot shows the Oracle Java SE Development Kit 8 Downloads page. A red circle highlights the 'Accept License Agreement' radio button, with a red callout bubble saying 'チェックを入れる' (Check it). A red line underlines the 'Mac OS X' row in the download table, with a red callout bubble saying 'Macの方はこちら' (Mac users click here). Another red line underlines the 'Windows x64' row, with a red callout bubble saying 'Windowsの方はこちら' (Windows users click here).

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.72 MB	jdk-8u91-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.69 MB	jdk-8u91-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.74 MB	jdk-8u91-linux-i586.rpm
Linux x86	174.92 MB	jdk-8u91-linux-i586.tar.gz
Linux x64	152.74 MB	jdk-8u91-linux-x64.rpm
Linux x64	172.97 MB	jdk-8u91-linux-x64.tar.gz
Mac OS X	227.29 MB	jdk-8u91-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.59 MB	jdk-8u91-solaris-sparcv9.tar.gz
Solaris SPARC 64-bit	98.95 MB	jdk-8u91-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.29 MB	jdk-8u91-solaris-x64.tar.gz
Solaris x64	96.78 MB	jdk-8u91-solaris-x64.tar.gz
Windows x86	182.29 MB	jdk-8u91-windows-i586.exe
Windows x64	187.4 MB	jdk-8u91-windows-x64.exe

②<https://netbeans.org/downloads/>

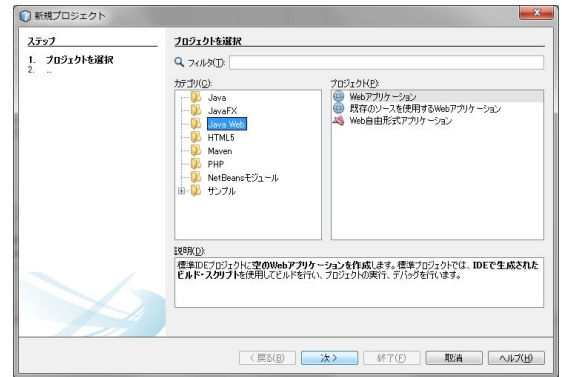
上記サイトより、『すべて』のインストーラーをダウンロードしてください。ダウンロード後は、インストーラーを起動しインストールを行ってください。

The screenshot shows the NetBeans IDE 8.1 download page. A red circle highlights the 'ダウンロード' (Download) button for the 'すべて' (All) package. A red line underlines the 'すべて' package row in the table.

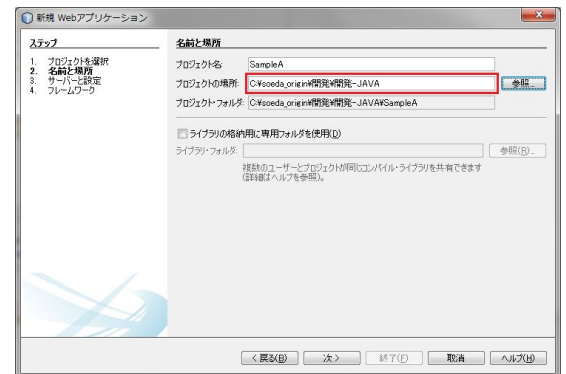
サポートテクノロジー *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	すべて
NetBeansプラットフォーム SDK	•	•				•
Java SE	•	•				•
Java FX	•	•				•
Java EE	•	•				•
Java ME		•				•
HTML5/JavaScript		•	•	•		•
PHP			•	•		•
C/C++					•	•
Groovy						•
Java Card(TM) 3 Connected						•
バンドル サーバー						•
GlassFish Server Open Source Edition 4.1.1		•				•
Apache Tomcat 8.0.27		•				•

③NetBeansの上部メニューから、
ファイル >> 新規プロジェクト
を開いてください

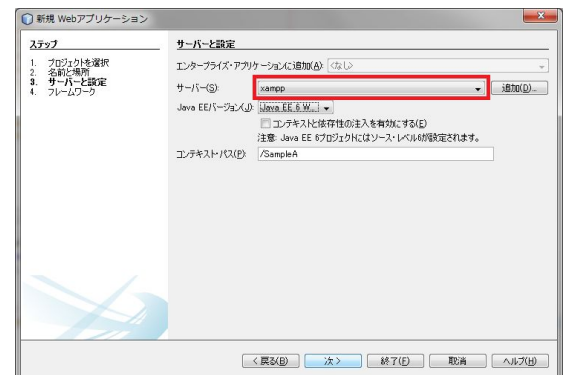
④カテゴリで『Java Web』を選択し、
プロジェクトで『Webアプリケーション』を選択してく
ださい。
できたら、『次>』を押して次へ。



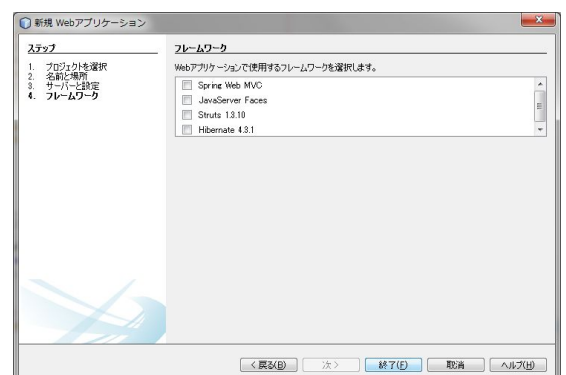
⑤プロジェクト名を適当に入力します。（「test」など）
できたら、『次>』を押してください。



⑥赤枠の部分がGlassFishになっているか確認し、
『次>』を押してください。



⑦最後のページは特に指定の必要はありません。



⑧ソースコードを書いて、ファイルを実行するとブラウザに表示されます。

【午前中の課題（チュートリアル式課題）】

「プログラミングって難しそう...」「実際どんな感じなの？」と思われる方も多いかもしれません。午後にお渡しする課題の前に、腕慣らしとして実際にプログラミングを動かしてみましょう！

◆課題：「Hello!」と5個表示してください。

ただし、「Hello!」を一つずつ表示させるのではなく、繰り返し構文を使って表示させてください。

手順：

1) 「メモ帳（テキストエディタなら何でも構いません）」を開いてください。

（Macの方はNetBeansを開いて、〔ファイル→新規プロジェクト→「JavaWeb」を選択して次へ→「camp」という名前を設定して次へ→完了〕と操作してください。プロジェクトが開かれるので、〔プロジェクト内の「webページ」を右クリック→新規→JSP→ファイル名を「camp.jsp」という名前で「終了」〕という流れで作成してください）

2) まずはJavaを書く準備をしましょう。下記をメモ帳に写経してください(ここにソースコードを書いていきます！という文字列は記述する必要はありません。改行だけしておきましょう)。

↓ここに半角スペースを入れてください

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    ここにソースコードを書いていきます！
%>
```

これでJavaのソースコードを動かす準備ができました。

「ここにソースコードを書いていきます！」というところに下記の通りソースコードを追加していきましょう。

3) ブラウザに文字を表示させてみましょう。

文字を表示するには、

```
out.print("表示させたい内容");
```

で表示することができます。

今回は「Hello!」と表示させたいので、

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    out.print("Hello!");
```

%>

と書きましょう。

4) 「Hello!」を5個表示させるために、繰り返し構文を使いましょう。
繰り返し構文は、

```
for(int i=初期値;i<繰り返す数; i++) {  
    out.print("処理内容");  
}
```

これで、「処理内容」を「初期値」から「繰り返す数」分だけ処理する、という事になります。

今回は「Hello!」と5個表示させたいので、

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<%  
    for(int i=0; i<5; i++) {  
        out.print("Hello!");  
    }  
%>
```

6) そのメモ帳を「xampp」→「tomcat」→「webapps」→「camp」フォルダ内に「**camp.jsp**」という名前で保存して下さい。
その際に、ファイルの種類を「すべてのファイル」にし、文字コードを「UTF-8」にしてください。

(Macの方は、上書き保存 [command+s])

7) xamppのTomcatがstartされていることを確認したら、ブラウザ（インターネットエクスプローラーかグーグルクローム）を開き、URLバーに「<http://localhost:8080/camp/camp.jsp>」と入力し、エンターを押してください。

(Macの方は、Netbeans上部の緑の右矢印ボタンをクリックしてスタート。URLは上記のように入力してください)

8) ブラウザに



Hello!Hello!Hello!Hello!Hello!

と表示されたら、成功です。（外枠は表示されません）

いかがでしょうか？

これでプログラミングの世界へ初めの一步を踏み出すことができました！

午後の課題では難しい課題に取り組んで頂きますので楽しみにしててください！

【午後の課題】

JAVA(JSP)を用いて、3の倍数でアホになったり5で犬になったりするプログラムを記述していただきます。

＜課題を解くコツ＞

- ❖ 「課題達成に必要なコーディング知識」を参考にしてみましょう！
 - 課題達成に必要なコーディング知識の一覧を参考にし、なるべくその部分を真似してやってみましょう。さすがにそのままコピーというわけにはいきませんが、周りを参考に置き換えができるはずです。

【課題】

あなたは、新人芸人「世界のナベアツ」のマネージャーです。デビューしたはいいものの、売れず飛ばず、まったくヒットする気配がありません。唯一ウケた芸はアホ顔でリアクションをとる芸だけ…。そこであなたはこんなネタを考えました「そうだ、1から20まで1ずつカウントして行って、3の倍数の時にだけアホになるネタなんてどうだろうか」。早速あなたは、プログラムでその動きを確認してみることにします。

課題1 1から20までカウントアップしながら、3の倍数の時にだけ[アホになる]という文字列を画面に表示するプログラムを作成してください。

解答例

```
1
2
3[アホになる]
4
5
6[アホになる]
7
.
```

「3の倍数でアホになる」ネタは、そこそこウケたものの、まだインパクトに欠けています。そこであなたはこう考えました「そうだ、3の倍数だけでなく、3のつく数字の時にアホになり、1から40までカウントアップするネタなんてどうだろうか」。

課題2 3を含む数字の時に、[アホになる]という文字列を画面に表示してください。また、カウントアップ回数を40まで増やしてください。

解答例

```
.  
11  
12[アホになる]  
13[アホになる]  
.
```

「3の倍数と、3のつく数字の時にだけアホになる」ネタは、空前の大ヒットとなりました。ナベアツはどのバラエティーにも引っ張りだこ。一躍売れっ子芸人です。この人気を維持するために、あなたはさらにこんなネタを思いつきました。「そうだ、これまでの内容に加えて、5の倍数で犬っぽくなるとかはどうか」。

課題3 5の倍数の時に、[犬っぽくなる]という文字列を出力してください。

解答例

```
.  
8  
9[アホになる]  
10[犬っぽくなる]  
11  
12[アホになる]  
13[アホになる]  
14  
15[アホになる][犬っぽくなる]  
.
```

課題4 [アホになる][犬っぽくなる]が同時に発生したとき、[アホ犬になる]が表示されるようにしてください。

解答例

```
.  
13[アホになる]  
14  
15[アホ犬になる]  
.
```

「3の倍数と、3のつく数字の時にだけアホになり、5の倍数で犬っぽくなる」ネタは大ウケしたものの、次第に飽きられ、以前のような人気ではなくなっていました。そこであなたは起死回生の一手として、こんなネタを思いつきました。「そうだ、アホになるたびにアホ度が上がり、犬になるたびに犬度も上がり、アホ犬になる時はそれまでのアホ度と犬度を合算したアホ犬度で表現し、カウントを4000までするのはどうか」。

課題5 [アホになる]、[犬っぽくなる]が出現したときにそれぞれの出現回数をカウントし、[アホになる:アホ度1]といった表示にしてください。[アホ犬になる]が出現したときはそれまでのアホ度犬度の合算を行い、[アホ犬になる:アホ犬度3]という表示にしてください。これを、4000までカウントアップしてください

解答例（下記の度数については、正しい数値とは限りません）

```
.  
.
400[犬っぽくなる:犬度81]
401
402[アホになる:アホ度238]
403[アホになる:アホ度239]
404
405[アホ犬になる:アホ犬度321]
406
.
```

<課題達成に必要なコーディング知識>

・変数と画面表示

4を画面に表示

```
out.print("4");
```

変数に格納してから表示

```
int suji = 4;
out.print(suji);
```

・条件分岐

もし変数の値が2の倍数ならば、「この値は偶数です」を表示

```
int suji = 8;
if(suji%2 == 0){
    out.print("この値は偶数です");
}
```

もし変数の値に6が含まれているならばその値を表示して改行

```
int suji = 652;
if(String.valueOf(suji).indexOf("6") != -1){
    out.print(suji);
    out.print("<br>");
}
```

変数の値に6が含まれる、または2の倍数ならば、「この数字は6が含まれ、偶数です」を表示

```
int suji = 64;
if(suji%2 == 0 || String.valueOf(suji).indexOf("6") != -1){
    out.print("この数字は6が含まれる、または偶数です");
}
```

変数の値に1が含まれる時には「この数字には1が含まれています」を、6が含まれるならば、「この数字には6が含まれています」を表示する

```
int suji = 11;
```

```

if(String.valueOf(suji).indexOf("1") != -1){
    out.print("この数字は1が含まれています");
}else if(String.valueOf(suji).indexOf("6") != -1){
    out.print("この数字には6が含まれています");
}

```

条件分岐の構造

```

if(条件1){
    条件1に合致した場合のみ実行される処理
}else if(条件2){
    条件1に合致せず、条件2に合致した場合のみ実行される処理
}else if(...){
    条件1に合致せず条件2にも合致せず....
}.....

```

複合条件の指定方法

|| または
&& かつ

・ 繰り返し処理

100回「こんにちは」と表示し、何回目かも表示する処理

```

int num = 1;
while(num<=100){

    out.print(num + "回目です");
    out.print("こんにちは<br>");

    num++;
}

```

1から50まで表示し、回数とそれが2で割り切れるときは[この値は偶数です]を表示する処理

```

int num = 1;
while(num<=50){
    out.print(num + "回目です");
    if(num%2 == 0){
        out.print("[この値は偶数です]");
    }
    out.print("<br>");
    num++;
}

```

繰り返し処理の構造

```

int カウント用の変数 = 1;
while(カウント用の変数 <= 繰り返す回数){
    繰り返し行いたい処理
    カウント用の変数++
}

```

【応用課題（WEB電卓）】

応用課題として、Javaで記述されたWebページ(Web電卓)の機能を改修してください。
修正対象のソースコードをお渡ししますので下記の通り課題に取り組んでください。

<課題を解くコツ>

- ❖ ソースコードの全文を読む必要はない！
 - このWeb電卓の課題では、一見大変な量のソースが書かれていますが、実は全文を読む必要はなく、日本語で「//一日体験課題」と書かれている部分の周辺をいろいろといじるだけで大丈夫です。
- ❖ 上下の記述を真似しよう！
 - 課題の部分の周辺には参考になる記述があります。なるべくその部分を真似してやってみましょう。さすがにそのままコピーというわけにはいきませんが、周りを参考に置き換えができるはずです。

<課題>

- 1.税込みボタン機能を追加する(例:100税込み → 108)
- 2.%ボタン機能を追加する(例:60% →0.6)
- 3.引き算の計算ができるようにする
- 4.値を0にし、初期状態にするためのボタン機能(C)を追加する(例:150C → 0)
- 5.数値が入力済みでなくとも+ボタンや税込み処理が動作してしまう(初期画面で+と押すと必ずエラーになる)。数値入力済みの時のみ動作するように条件分岐させる
- 6.「.(小数点)」ボタンと、その機能を追加する