

情報処理及び実習 第3回(4/23)

月曜日4,5-1時限(第1実習室)
火曜日2時限(第4実習室)4/17,4/24
鈴木良弥, 丹沢勉
TA: 小佐野, 佐藤

授業資料 <http://www.ccn.yamanashi.ac.jp/~ysuzuki/JMinfo/index.html>

前回の復習

- 電子メールの使い方

電子メールの使い方

- 統合メール
- 学籍番号t12jm001なら
- t12jm001@yamanashi.ac.jp

WebMailの使い方

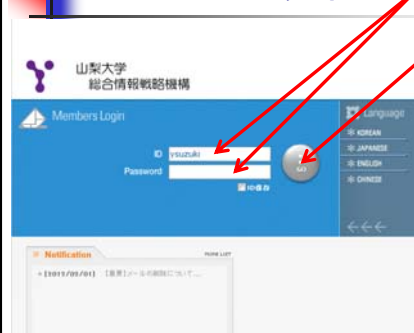
- ログイン
- 受信トレイの確認
- メールの作成
- 送信
- 受信
- ログアウト

WebMail ログインページへ

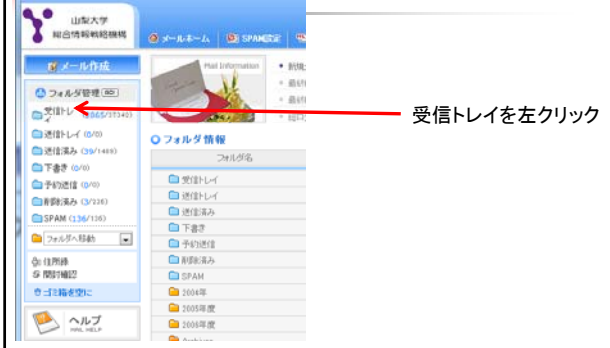
- Webブラウザ(IEまたはFirefox)を開く
- <https://mailsv.yamanashi.ac.jp/cgi-bin/index.cgi> をブックマークしておく
- [山梨大学](http://www.yamanashi.ac.jp/)(<http://www.yamanashi.ac.jp/>)
- →総合情報戦略機構→WebMailログインページ

WebMail ログイン

ログインID,
Passwordを入力
マウスを左クリック



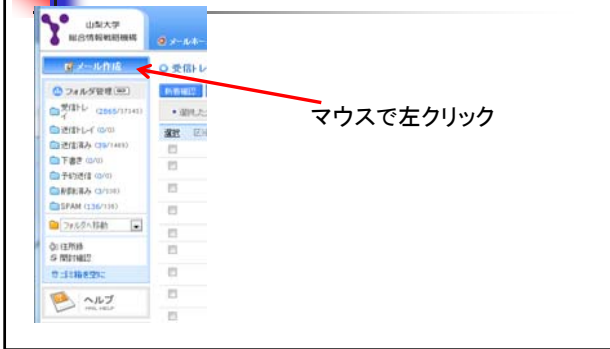
WebMail 受信トレイの確認



WebMail 受信トレイ内のメールを読む

- メールの件名を左クリック

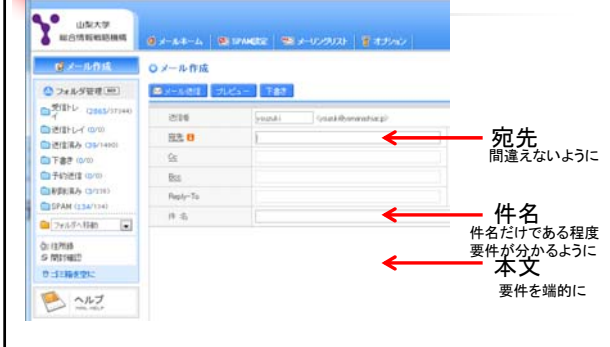
WebMail メールの作成



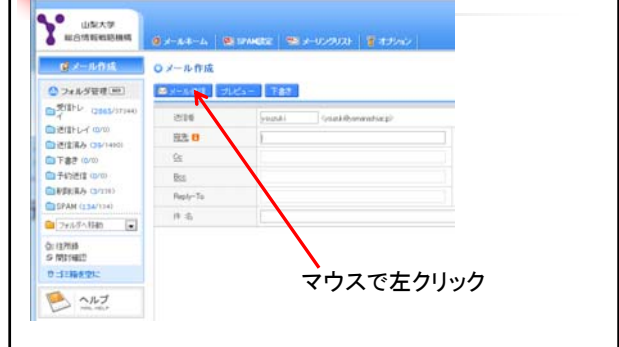
メール作成時の注意

- 件名はわかりやすく（件名だけである程度要件がわかるように）
- 本文もわかりやすくなるよう工夫
- 読み直してから送信
 - 誤字, 脱字は無いかな？
 - 読み易いかな？
 - 曖昧さはないかな？
 - 失礼ではないかな？

メール作成



WebMail 送信



WebMail 受信

- まだ受信トレイを開いていないなら
 - 「受信トレイ」を押す
- すでに受信箱を開いているなら
 - →「新着確認」を押す
 - →「受信トレイ」を押す
- 新着メールを見つけて件名を押す
- メールの内容を確認

WebMail 返信

- メールの内容を確認
- メールの上の返信ボタンを押す
- 「---- Original Message -----」より上に返信内容を書く
- 件名は(普通は)修正しない
- メールの上にあるメール送信ボタンを押す

実習(メール送信, 受信)

- 自分の次の学籍番号の人にメールを送信
 - T10MI042→T12JM001
 - T12JM001→T12JM002
 - T12JM061→T10MI042
- 内容は「フットサルサークルへの勧誘」
- 内容が気に入らなければ違う内容でもOK
- メールが来たら返信

前回の宿題

- ~~CNSで学内の友達にメッセージを送る~~
- Webメールで自分の携帯にメールを送る
 - 携帯を持っていない場合はPCメールのアドレスでも大学の統合メールアドレスでもよい
- Gold Finger Schoolの「日々の練習」を1回ずつTry!

今日の実習 C言語プログラミング入門の準備

- Visual Studioの起動
- コンソールアプリの作り方
- C言語でのプログラミング
 - プログラミング入門ではないので、サンプルプログラムが理解できなくても良い

Visual Studioの使い方 (プログラミング入門で使用)

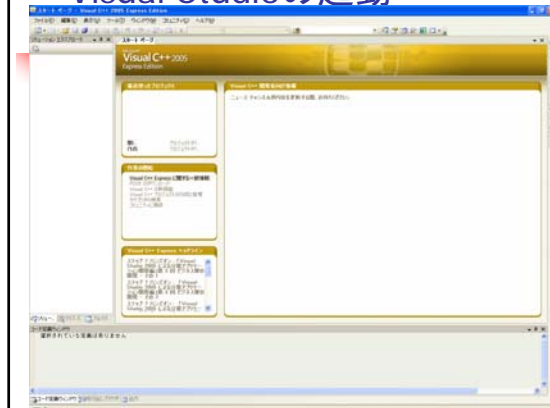
- Visual Studio
 - Microsoftのプログラミング用統合環境
 - C++(C), C#, Basicなどの複数言語で利用可能
 - プログラミングに必要な機能が満載
 - レイアウト作成
 - コード補間機能(IntelliSense)
 - コンパイラ
 - リンカ
 - デバッガ

Visual Studioの起動

- 授業で使用するVisual Studio:
 - Visual Studio 2005 Express
- スタート→すべてのプログラム→Visual C++ 2005 Express Edition→Microsoft Visual C++ 2005 Express Edition

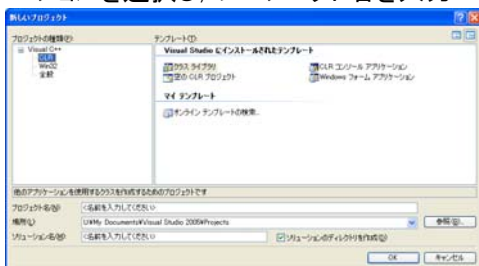


Visual Studioの起動



プロジェクトの作成

- ファイル→新規作成→プロジェクト
- プロジェクトの種類でWin32を選び、テンプレートとして「Win32コンソールアプリケーション」を選択し、プロジェクト名を入力



ソースコード入力まで

- 追加のオプション: 空のオプション
- 「完了」ボタンを押す
- ソリューションエクスプローラのソースファイルを右クリック→追加→新しい項目
- 新しい項目: C++ファイル(.cpp)を選択
- 下のファイル名入力欄に「~.c」と入力
- ソースコードを入力

ビルド, 実行

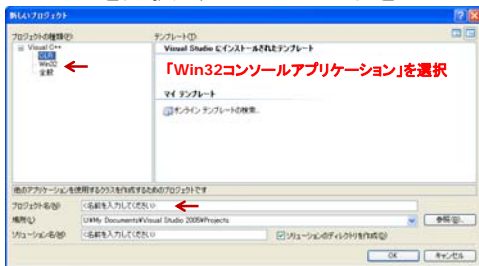
- メニューからビルド→ソリューションのビルド
- エラーが無い確認
- デバッグ→デバッグの開始
- 画面がすぐに消えてしまう場合はブレイクポイントを追加
- デバッグ→デバッグの停止

実習1: プロジェクト hello の実行

- Hello, world を表示させる

プロジェクト hello の作成 1/5

- ファイル→新規作成→プロジェクト
- プロジェクトの種類でWin32を選び、テンプレートとして「Win32コンソールアプリケーション」を選択し、プロジェクト名を入力

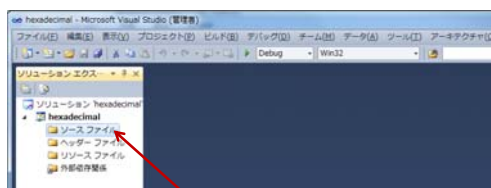


プロジェクト hello の作成 2/5

- 追加のオプション:空のオプション
- 「完了」ボタンを押す

プロジェクト hello の作成 3/5

- ソリューションエクスプローラのソースファイルを右クリック→追加→新しい項目



右クリック

プロジェクト hello の作成 4/5

- 新しい項目 :C++ファイル(.cpp)を選択

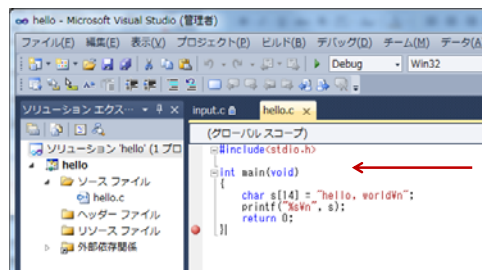


- 下のファイル名入力欄に「hello.c」と入力



プロジェクト hello の作成 5/5

- ソースコードを打ち込む



ソースコード hello.c

プログラム1

```
#include<stdio.h>
```

```
int main(void)
{
    printf(\"hello, world\\n\");
    return 0;
}
```


ブレークポイント(一旦停止) グレー部分を左クリック

ソースコード hello.c プログラム1

```
#include<stdio.h> 入出力用ヘッダファイルの組み込み
```

```
int main(void) main関数
```

```
{  
    フォーマット付き出力  
    printf("hello, world¥n");  
    return 0;  
    返り値  
}
```

 ブレークポイント(一旦停止) グレー部分を左クリック

実行結果

- 「hello, world」が一行目に表示されたウィンドウが表示される
- 終了するには デバッガー→デバッグの停止

作成したプロジェクトの格納場所

- プロジェクト(自分で変更しなければ)
 - ドキュメント ▶ Visual Studio 2005 ▶ Projects ▶ hello
- ソースファイル(hello.c)
 - hello ▶ hello ▶ hello.c

プログラム 1e-1

```
#include<stdio.h>
```


このセミicolon「;」を消してビルドするとどうなるか確認

```
int main(void)  
{  
    printf("Hello World¥n");  
    return 0;  
}
```

プログラム 1e-2

```
#include<stdio.h>
```


```
int main(void)  
{  
    printf("Hello World¥n");  
    return 0;  
}
```

 この「}」を消してビルドするとどうなるか確認

プログラム 1e-3

```
#include<stdio.h>
```

```
int main(void)  
{  
    printf("Hello World¥n");  
    return 0;  
}
```

 注釈マーク: このマーク以降はビルドに無視される
行頭に//をつけてビルドするとどうなるか確認

ビルド時に下のウィンドウに表示されるメッセージは重要

- 1:ビルドに成功しました。
- 1e-1:構文エラー: ';' が 'return' の前にありません。
- 1e-2:左側 中かっこ '{' に対応するものが '...' で見つかる前に EOF が検出されました。
- 1e-3:左側 中かっこ '{' に対応するものが '...' で見つかる前に EOF が検出されました。

プログラム 1m-1

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
char s[14] = "hello, world";
```

```
printf("%s\n", s);
```

```
return 0;
```

```
}
```

枠内を書き換える

変数(文字配列)s

```
char s[14] = "hello, world";
```

0	1	2	3	4	5	6	7	8	9	10	11	12	
h	e	l	l	o	,		w	o	r	l	d	¥0	

ヌル文字
文字列の終端

プログラム 1m-2

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
char s[14] = "hello, world!";
```

```
printf("%s\n", s);
```

```
return 0;
```

```
}
```

「!」を1つ追加して
ビルド→デバッグ開始

0	1	2	3	4	5	6	7	8	9	10	11	12	13
h	e	l	l	o	,		w	o	r	l	d	!	¥0

プログラム 1m-3

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
char s[14] = "hello, world!!!";
```

```
printf("%s\n", s);
```

```
return 0;
```

```
}
```

「!!」を追加して
ビルド→デバッグ開始

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
h	e	l	l	o	,		w	o	r	l	d	!	!	¥0

配列のオーバーランに注意

- コンパイラは配列の範囲チェックをしてくれない
- 配列のオーバーランによる影響は予測不可能
- プログラマが責任をもってチェック

ソリューション(プロジェクト)2 四則演算 arithmetic.c

arithmetic.c (少し問題のあるプログラム)

```
#include<stdio.h>
int main(void)
{
    int a, b;
    printf("a="); scanf("%d", &a);
    printf("b="); scanf("%d", &b);
    printf("%d+%d=%d\n", a, b, a+b);
    printf("%d-%d=%d\n", a, b, a-b);
    printf("%d*%d=%d\n", a, b, a*b);
    printf("%d/%d=%f\n", a, b, (float)a/b);
    return 0;
}
```

実習2

- プログラムを打ち込んで、実行する。
- a=5, b=3を入力
- a=3, b=10
- a=a, b=b
- printf("%d/%d=%f\n", a, b, (float)a/b); → printf("%d/%d=%f\n", a, b, a/b);

実習3

- 10進数, 16進数, 2進数表記を並べて表示
- プロジェクト名: hexadecimal
- ソースファイル名: hexadecimal.c
- 次のページのプログラムを打ち込んだ後、いろいろ変更してどんなエラーや結果が表示されるかを観察

hexadecimal.c

```
#include<stdio.h>
#include<stdlib.h>
#define BUFF_SIZE 10

int main(void)
{
    char buf[BUFF_SIZE];
    int i;
    printf("10進数 16進数 2進数\n");
    for(i=0; i<=32; i++)
    {
        printf(" %02d %02x %08s\n", i, i, itoa(i,buf,2));
    }
    return 0;
}
```

明日の授業に持ってくるもの

- プログラミング入門の教科書
 - 教科書の例題を打ち込んで実行する
- USBメモリ
 - 自分が作った(テキスト)ファイルをUSBメモリにコピーする



今回の宿題

- 実習1～3を実際に打ち込んで動作を確認
- 自分の作成したソースファイルがどこに格納されるかを確認
- プログラムを少し変更してエラーや表示の違いを観察する
- Gold Finger Schoolの「日々の練習」を1回ずつTry!