プログラミング論I
(1)C言語の入門

電子情報工学専攻　日浦 慎作

座席は自由です

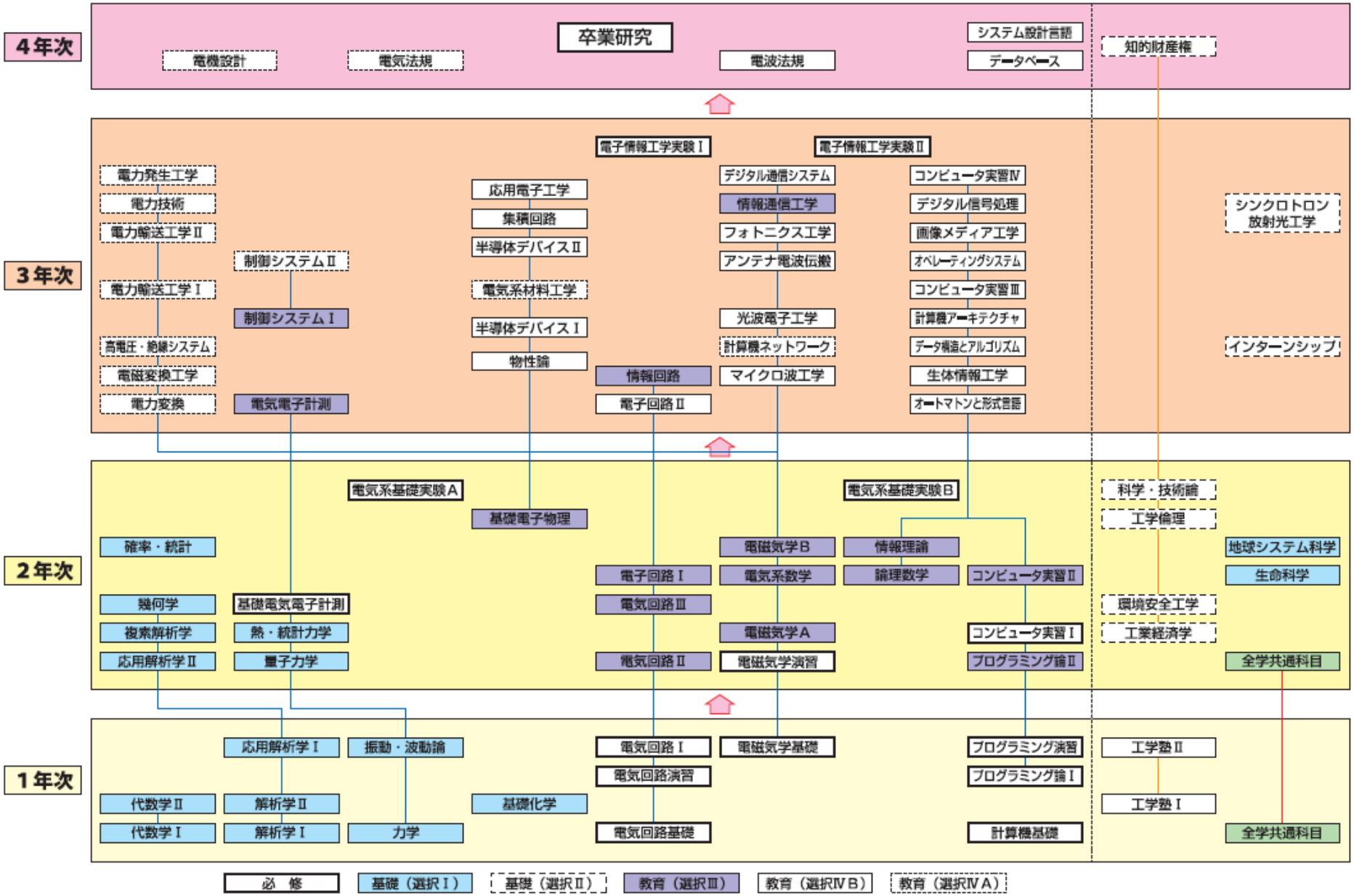
最初にアナウンス

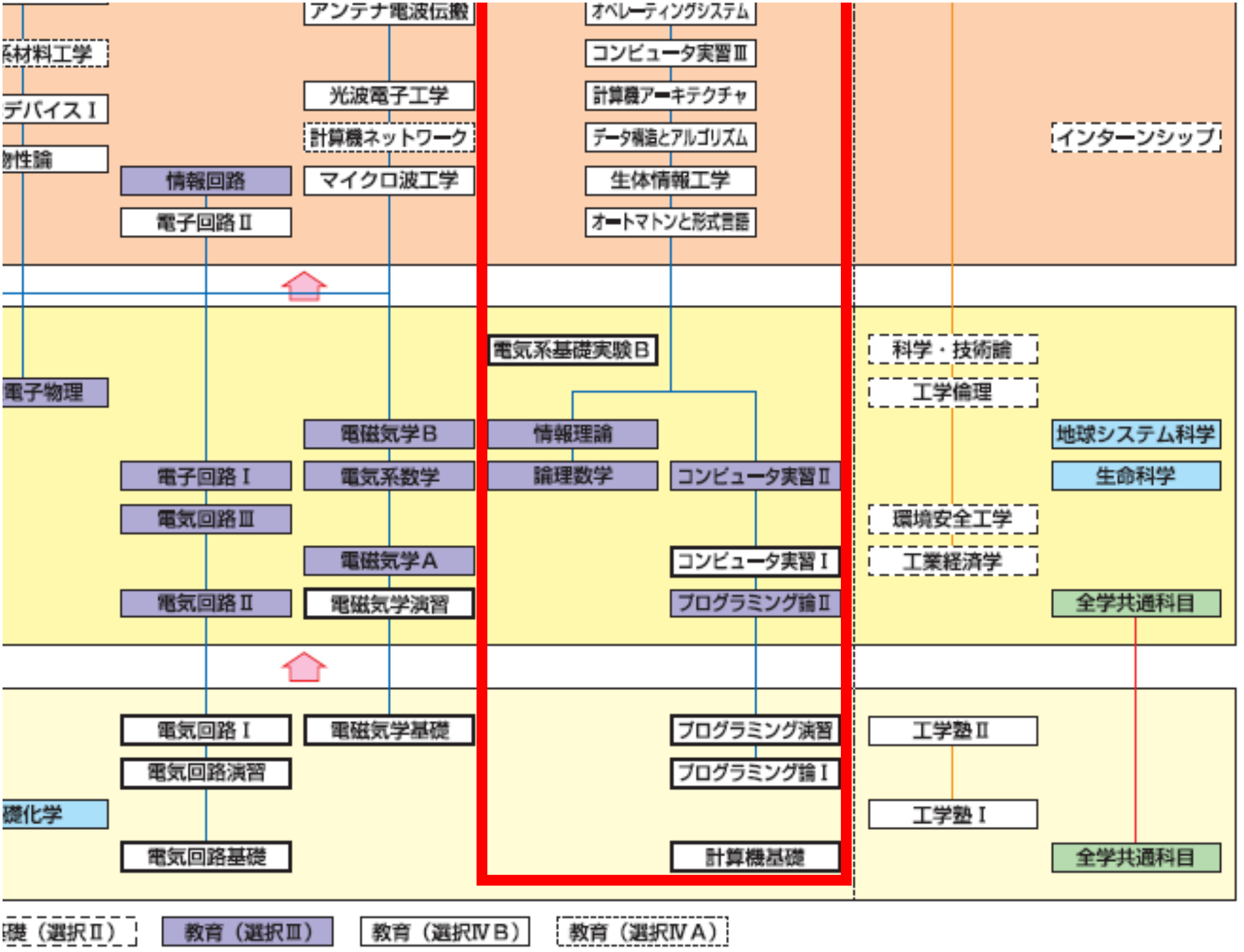
- 木曜１限「プログラミング演習」部屋割
  - 7201教室：TB19L001～TB19L067
  - 7204教室：TB19L068～，過年度生

  遅れないように！！

カリキュラム・マップ

工学部ホームページ
科目履修系統図



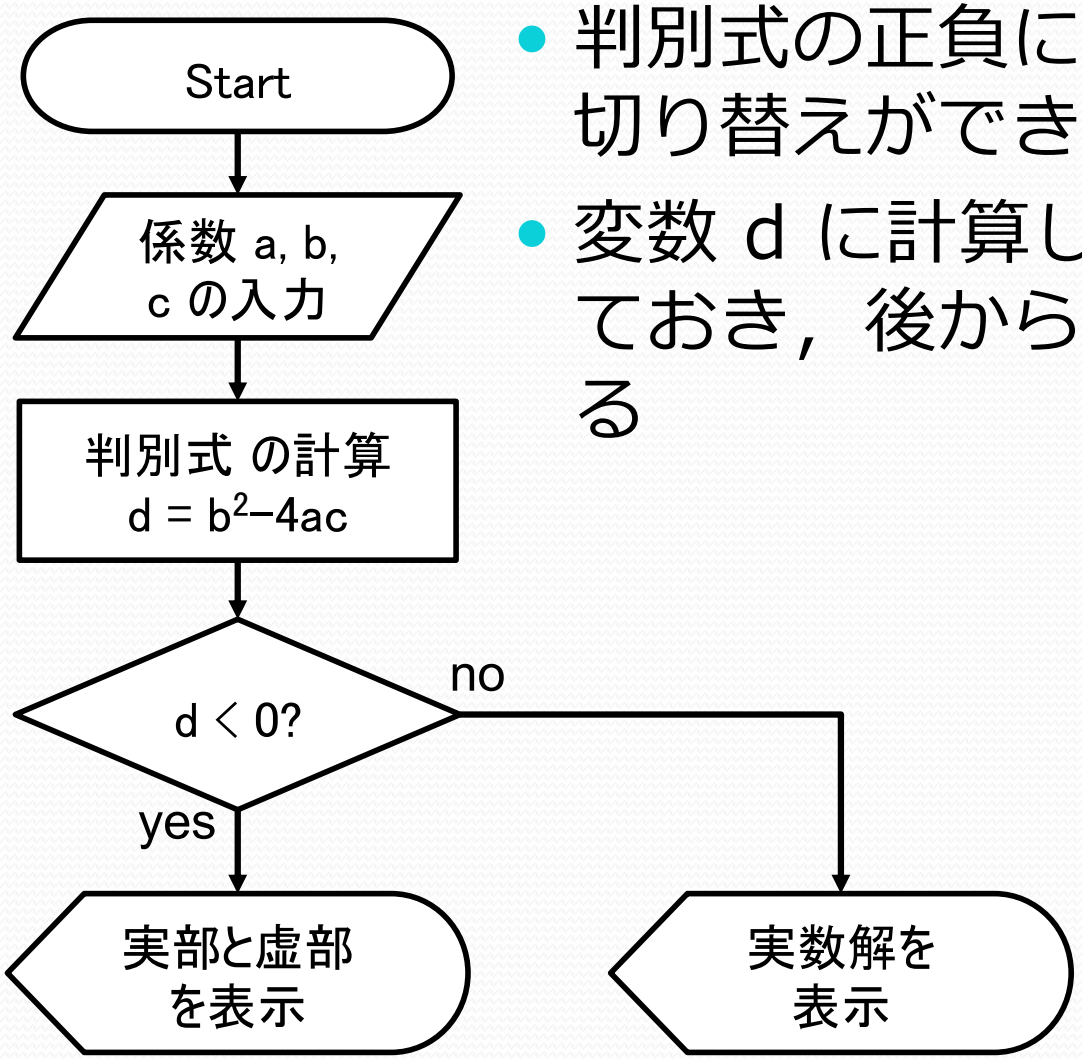

アンテナ電波伝搬
オペレーティングシステム
系材料工学
コンピュータ実習Ⅲ
光波電子工学
計算機アーキテクチャ
デバイスⅠ
計算機ネットワーク
データ構造とアルゴリズム
インターンシップ
物性論
情報回路
マイクロ波工学
生体情報工学
電子回路Ⅱ
オートマトンと形式言語
電気系基礎実験B
科学・技術論
電子物理
工学倫理
電磁気学B
情報理論
地球システム科学
電子回路Ⅰ
電気系数学
論理数学
コンピュータ実習Ⅱ
生命科学
電気回路Ⅲ
環境安全工学
電磁気学A
コンピュータ実習Ⅰ
工業経済学
電気回路Ⅱ
電磁気学演習
プログラミング論Ⅱ
全学共通科目
電気回路Ⅰ
電磁気学基礎
プログラミング演習
工学塾Ⅱ
電気回路演習
プログラミング論Ⅰ
礎化学
工学塾Ⅰ
電気回路基礎
計算機基礎
全学共通科目
礎（選択Ⅱ）
教育（選択Ⅲ）
教育（選択ⅣB）
教育（選択ⅣA）

プログラミング

- 計算機が直接実行出来るのは機械語である
  - 機械語で直接，プログラムを作るのは難しい.
  - 機械語は，機種によって異なる．作り直すのが大変.
- プログラム言語とは？
  - Ｃ言語など．機械語よりもずっと分かりやすい.
    - そのため高水準言語，高級言語などとも呼ばれる.
  - 計算機によって，機械語に翻訳してから実行する.
    - 機種によって書き直す必要が少なくなる（移植性という）
- 何ができるのか？
  - 計算ができる（あたりまえ）
  - 計算の手順を決めて，自動化できる
    - 条件判断（場合分け）や，繰り返し処理などができる

プログラムの例

例：２次方程式 $ax^2+bx+c = 0$ の解を求める



- 判別式の正負による計算手順の切り替えができる
- 変数 d に計算した値を代入しておき，後から使うことができる

C言語とは（教科書p.2）

- 1971～2年，「大きなプロジェクト」ではなく，ある「天才」により設計される
  - 設計者：デニス・M・リッチー
- UNIX オペレーティングシステムとともに発展
  - UNIX (Linux 等）はほぼ全てC言語で記述されている
- 現在，C言語が利用できない環境（計算機）は，ほとんどない
  - 大変多くのソフトウェア製作に利用されている
  - 家電製品の制御などにも利用される

C言語の普及と発展

- 開発者らによる教科書
  - K&R (Kernighan & Ritchie)
- 標準規格化（1989年）
  - ANSI-C（米国規格協会）
- 他の言語への影響
  - Java やC++, C# などでも，基礎的な構文や記号の使い方はC言語に倣ったものが多い

C言語をマスターすれば，他の言語の習得も楽！

C言語と機械語の例

C言語：

```
a = 10;
b = 20;
c = a + b;
```

機械語（アセンブラ：Intel 社の CPU の場合）

```
movl $10, -12(%ebp)
movl $20, -16(%ebp)
movl -16(%ebp), %eax
addl  -12(%ebp), %eax
movl %eax, -20(%ebp)
```

プログラム言語の翻訳

- コンパイラ
  - プログラムを実行するより前に，
    プログラム全体を翻訳してしまう方式.
  - 皆さんがプログラミング演習で使うもの.
  - 利点：実行速度が速い.
    事前にプログラミングのミスをチェックできる.
  - C言語はコンパイラ言語.
- インタプリタ
  - プログラムを，それぞれの行の実行の直前に，
    その都度翻訳しながら実行していく方式.
  - BASIC のほか，最近は JavaScript や Perl, PHP などのweb関連やスクリプト言語で多く用いられている.
  - 利点：すぐに試すことが出来る.
    欠点：実行速度が遅い.

プログラミング

1. プログラム仕様の作成
   - プログラムの機能，満たすべき性能などを決める．
2. 設計
   - アルゴリズムや処理手順を検討して決定する．
3. コーディング
   - プログラム言語として記述していく．
4. デバッグ
   - プログラムに不具合がないか調べ，適宜修正する．

「バグ」とは？

- プログラムの不具合．語源は「虫」．

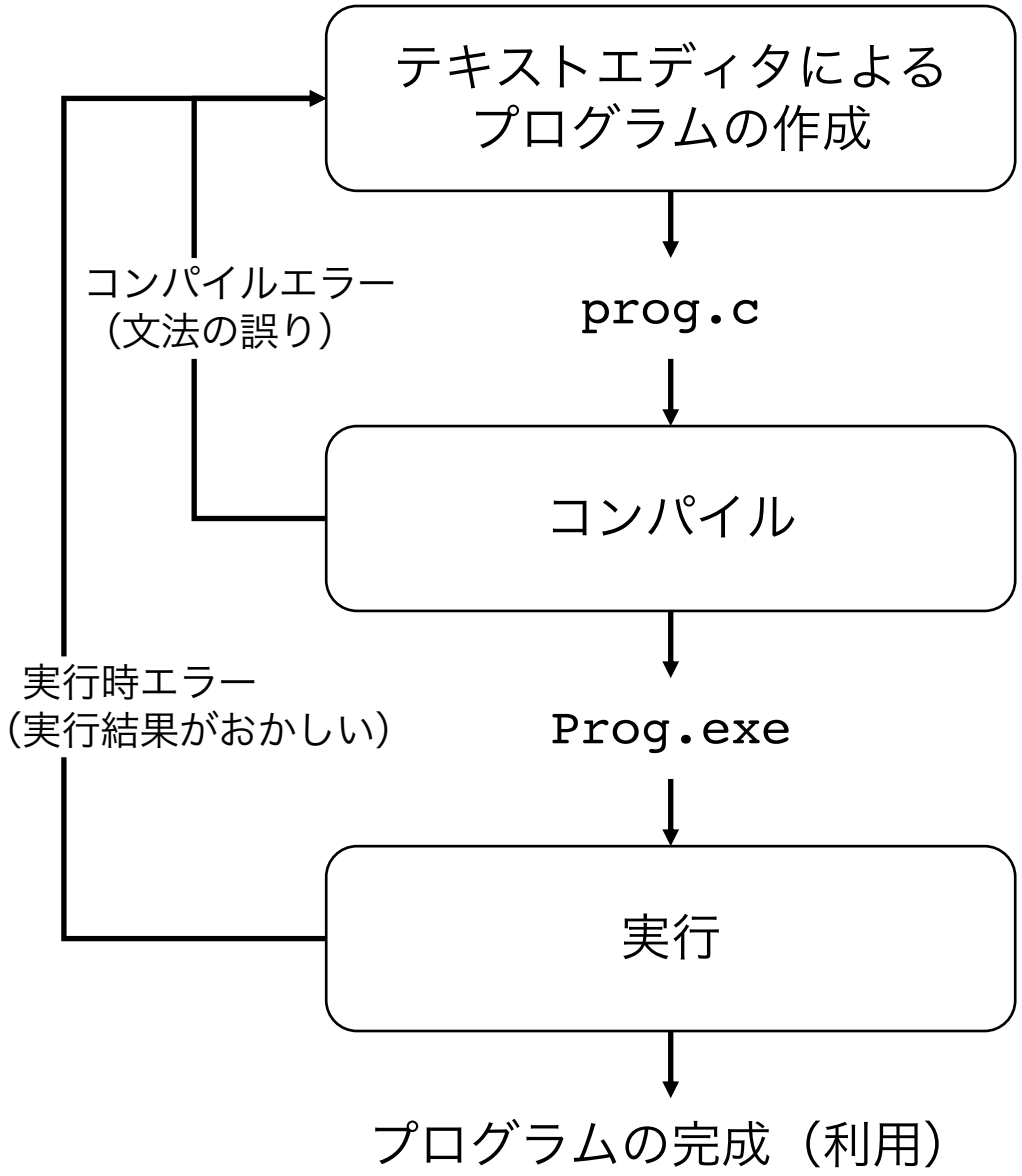
プログラム言語の種類(1)

- C言語
  - 多くのソフトウェアがＣ言語で作られている.
  - オペレーティングシステムもほとんどがＣ言語で作られている.
- COBOL
  - 事務処理，会計処理向け言語．銀行の大型システム等.
- BASIC
  - 初心者向け言語．普通，インタプリタで実行する.
- Fortran
  - 科学技術計算（行列計算など）向け言語.
- PASCAL
  - 大学の研究者が作った．教育向け.

プログラム言語の種類(2)

- オブジェクト指向
  - 「データ構造」（対象．オブジェクト．）と，
    それに対する操作を組にして記述していく方法．
  - 「手順」よりも「対象」に注目した考え方．
- C++
  - C言語を元にオブジェクト指向を取り入れた言語．
- Java
  - C言語より機種依存性が低く，高機能．
  - 携帯電話のアプリ制作などでも広く使われていた．

プログラム作成・検証の流れ



・サクラエディタなどを用いる
・テキストファイルとして保存する

・○○.c というファイル名にする

・コマンドプロンプトで以下を実行：
`cl prog.c`

・実行可能ファイルが作成される
(Windows の場合は ○○.exe)

・コマンドラインで以下を実行
`prog.exe`

最初のC言語のプログラム

```
#include <stdio.h>
int main(void)
{
   printf("Hello World.¥n");
   return 0;
}
```

printf などを使えるようにする

プログラムはここから始まりますよ，という意味

この括弧の間にプログラムを書く

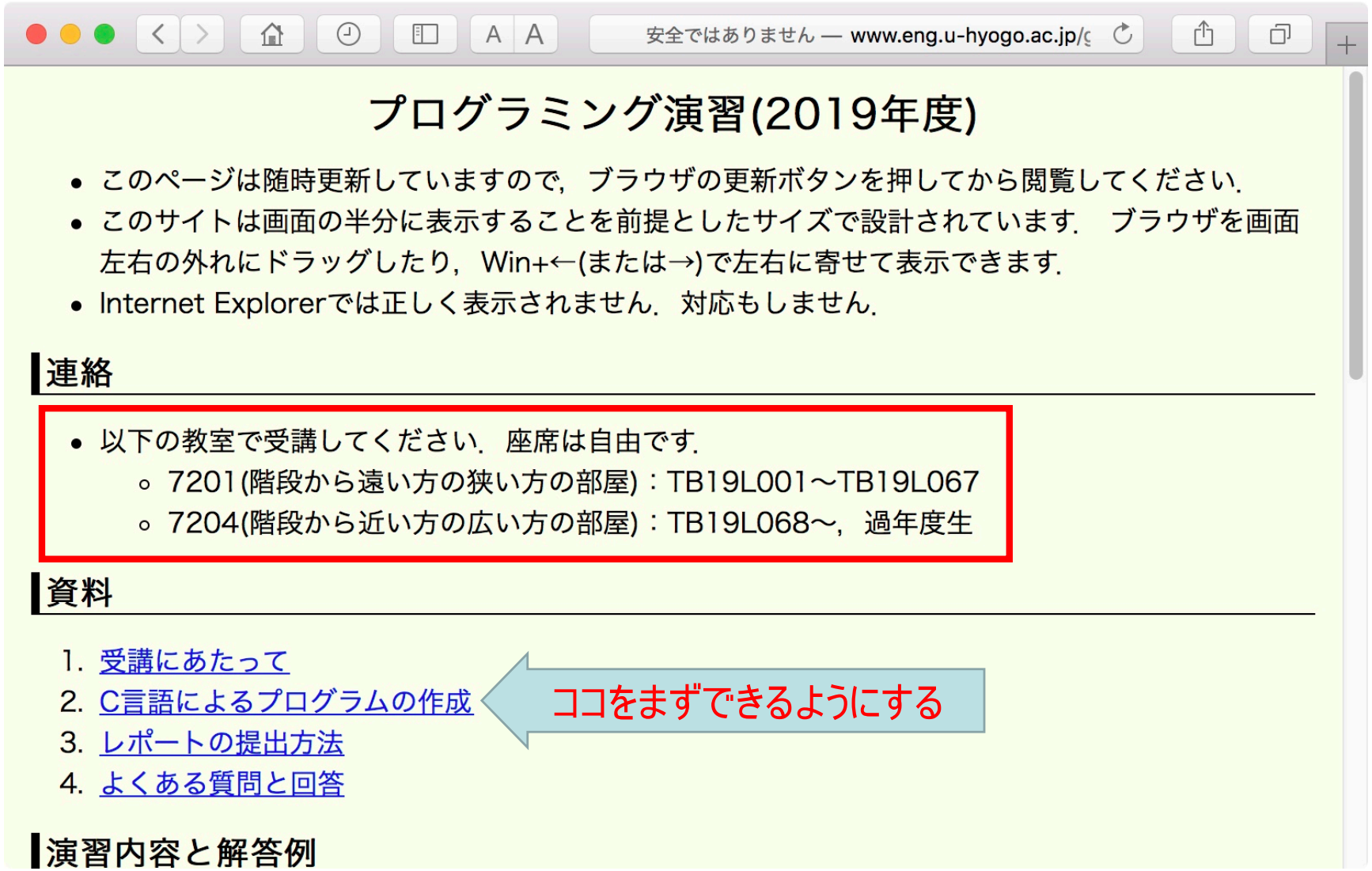
Hello World. と出力し，改行

プログラムを正常に終了させるための一文

難解な（意味不明の）部分が多いと思いますが，これはおいおい説明します．

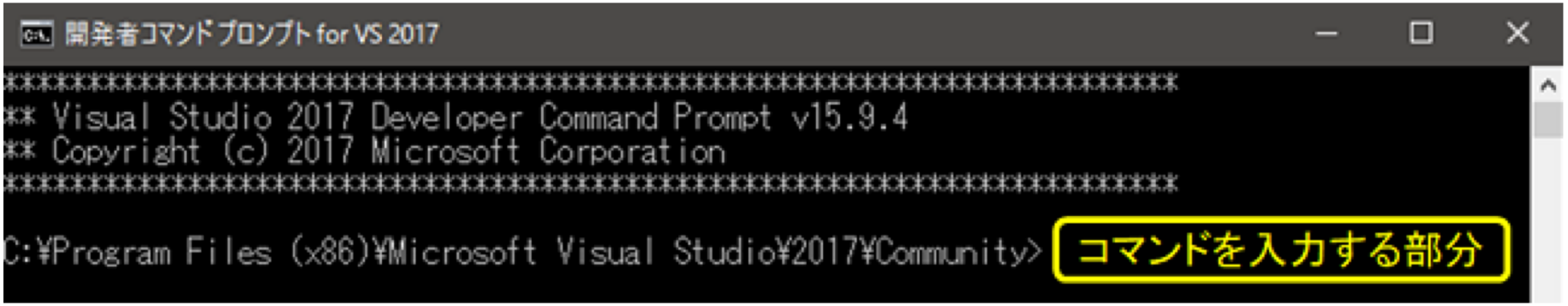
プログラミング演習のホームページ

http://www.eng.u-hyogo.ac.jp/group/group30/pe/

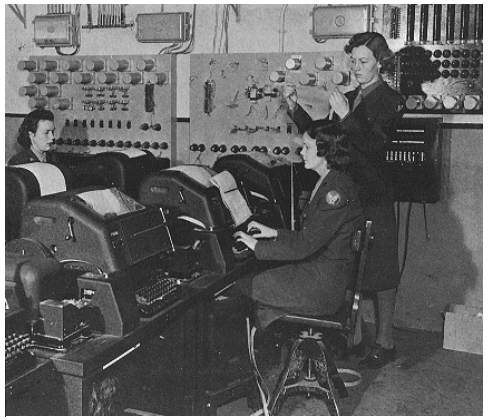


コマンドプロンプトってなんだ？

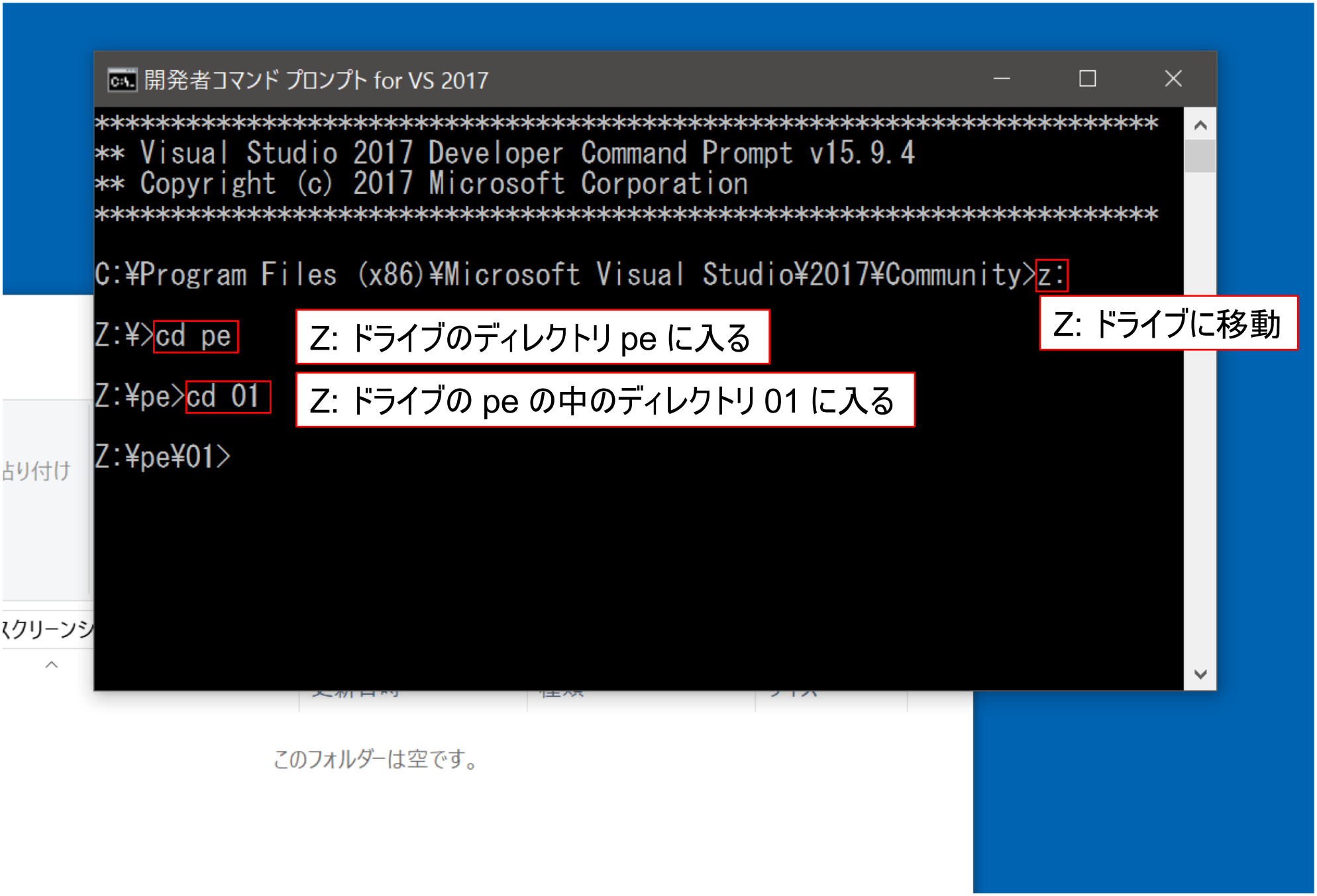
- 文字の入出力だけでコンピュータを操作するもの



- 歴史的経緯
  - コンピュータ登場前，遠隔地間の文書通信手段として，テレタイプという装置が使われていた
    - 電動式のタイプライターを遠隔地間で接続したもので，A地点で入力した文字がB地点で印刷されるようなもの
  - 第2次世界大戦後，コンピュータが出現
    - コンピュータに「テレタイプ端末」を接続して文字の入力や結果の出力ができるようにした
    - 表示方式がプリンタから画面表示に変わっても，変わらずその文化が残り使い続けられている
    - 標準入出力として使われる（後に講義）

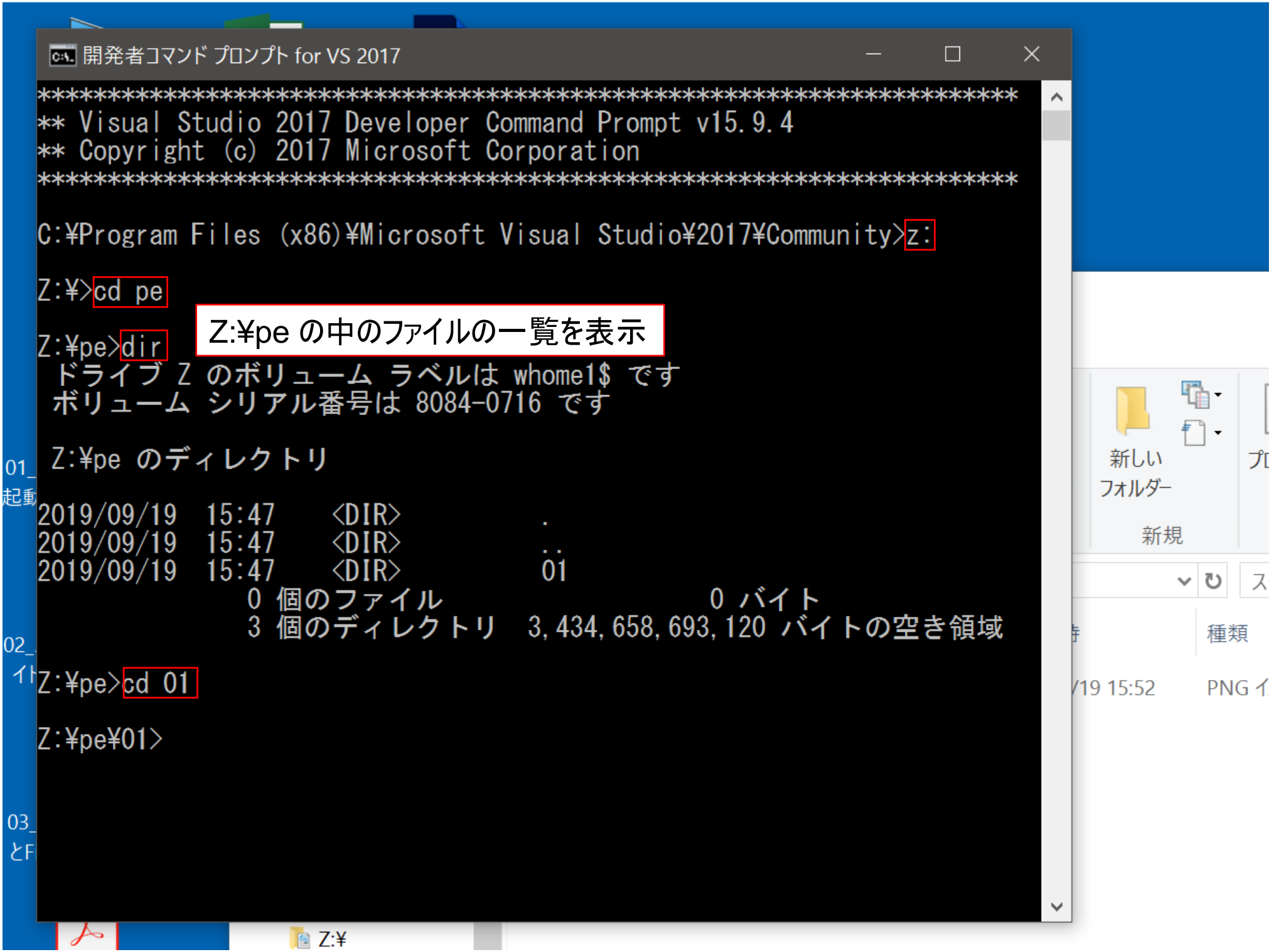


赤枠内は自分がキーボードから入力した文字



コマンドプロンプトについて

- Windows のファイル操作コマンド
  - ファイルの一覧を表示　・・ dir
  - ディレクトリ（フォルダ）の移動　・・ cd
    - cd 01　とすると，01 の中に入る
    - cd ..　とすると，１段階さかのぼる（戻る）
  - フォルダの作成　・・ mkdir
    - mkdir 02　とすると，フォルダ 02 ができる
  - コンパイル ・・　cl
    - cl prog.c とすると，prog.exe ができる
  - 他にも，copy, del, move, rename など．．


開発者コマンド プロンプト for VS 2017
********** Visual Studio 2017 Developer Command Prompt v15.9.4
** Copyright (c) 2017 Microsoft Corporation
****
C:¥Program Files (x86)¥Microsoft Visual Studio¥2017¥Community>z:
Z:¥>cd pe
Z:¥pe>dir
Z:¥pe の中のファイルの一覧を表示
ドライブ Z のボリューム ラベルは whome1$ です
ボリューム シリアル番号は 8084-0716 です
Z:¥pe のディレクトリ
2019/09/19 15:47 <DIR> .
2019/09/19 15:47 <DIR> ..
2019/09/19 15:47 <DIR> 01
0 個のファイル 0 バイト
3 個のディレクトリ 3,434,658,693,120 バイトの空き領域
Z:¥pe>cd 01
Z:¥pe¥01>
新しいフォルダー
新規
種類
Z:¥

C言語のプログラムの構造

```
#include <stdio.h>
int main(void)
{
   int seisu;
   seisu = 5;
   printf("seisuの値は%dです¥n",seisu);
   return 0;
}
```

変数（値を入れるための名前）の準備

変数定義が先

変数に値を入れる

値の表示

実行部分は後

それぞれの文は “;” で終わる（区切られる）.

演算子について

- 加減乗除

  a + b　　a - b　　a * b　　a / b

- 剰余（割った余り）

  a % b　　a を b で割った余り（整数のみ）

- 優先順位

  数学と同じで * / のほうが + - よりも優先順位が高い

  ( ) を使うことができる（不安なときは使うべき）

- べき乗（a^b）の演算子はありません

代入について

- C言語の = は左側に書いた変数への「代入」
  - 右辺と左辺が後々まで等しいと決めるわけではない

```
int a, b;
a = 5;
b = a;
a = 3;
```

とすると，bの値は5のまま（3にはならない）

  - 左辺には数式を書くことはできない

```
int a;
a + 5 = 7;
```

とかいう書き方はできない（エラーになる）

printf の使い方

- 文字列を表示

  printf(“Hello World!”);

バックスラッシュ ＼ は

¥ で入れる（¥ で表示されることも，＼で表示されることもある）

- 改行などの特殊文字も利用可能

  （¥n は改行を意味する）

  printf(“Hello World!¥n”);

- 数値を表示

  printf(“5 x 6 = %d¥n”, 30);

  - 出力結果は　5 x 6 = 30 となる（最後に改行）

  printf(“result = %d¥n”, hensu);

  - hensu = -7 のとき，出力結果は　result = -7

C の文法

- 文 ・・・ ;（セミコロン）で終わる．処理実行の単位
  - seisu = 5;　　・・・ 変数 seisu に 5 を代入
  - printf(“result = %d¥n”, hensu);
    ・・・printf により画面表示する
- 識別子　・・・　変数，関数などの名前
  - 上の例では seisu, printf, hensu が該当
  - 自分で命名できる（すでに用意されているものもある）
- 文字列　・・・　“ と “ でくくられた文字
  - コンパイラは文字列を解釈しない．定数 5 などと同等
- 関数　　・・・　識別子(…) の形のもの

文法とプログラムの構造

```
#include <stdio.h>
int main(void) {
    int seisu;

    seisu = 5;
    printf("seisuの値は%dです¥n", seisu);
    return 0;
}
```

行頭が#はプリプロセッサ（特別扱い）

文字列
識別子
予約語
凡例

予約語一覧

auto	const	double	float	int	short	struct	unsigned
break	continue	else	for	long	signed	switch	void
case	default	enum	goto	register	sizeof	typedef	volatile
char	do	extern	if	return	static	union	while

識別子は，アルファベットか数字で作る．ただし↑の予約語は使えない
（ただし先頭はアルファベットのみ）