

情報I,IIで扱う プログラミングへの期待

大阪電気通信大学 兼宗進

小学校でのプログラミング

- 教科での利用(算数/理科など)
- プログラミングを体験(総合的な学習の時間)

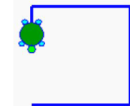
- 言語、教材の指定はなし
- 全学年の全教科で利用を推奨
(学校ごとのカリキュラムマネジメント)

小中高の連続したプログラミングの流れ(小学校)

- **小学校: 楽しさを体験する、嫌いにしない**
 - 教科で手順(順次/反復/分岐)を活用
- 中学校: 世の仕組みを知る
 - 計測・制御(あらゆる機器で利用)
 - 双方向コンテンツ(アプリは通信)
- 高校: 将来はセンター試験科目?
 - 情報I: アルゴリズム、シミュレーション
 - 情報II: データサイエンス、情報システムとプログラミング
-
- **補足: 情報活用能力も重要(特に文字入力)**

プログラミングで教科を学習

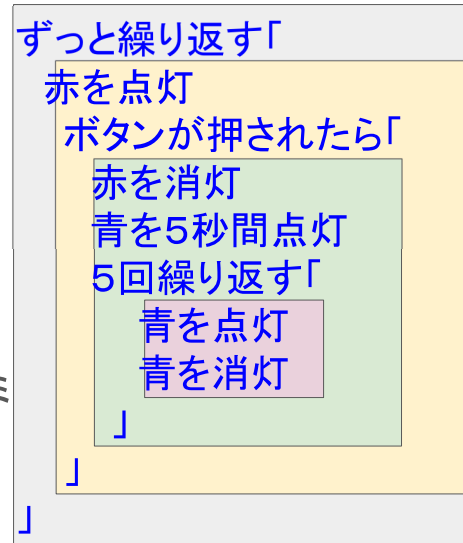
- 例示: 小5算数「正多角形」など
- 私が公開しているドリトル言語の例



**かめた=タートル！作る(青)線の色。
「
かめた！100 歩く 90 左回り。
」！4 繰り返す。**

小学校の市販教材例

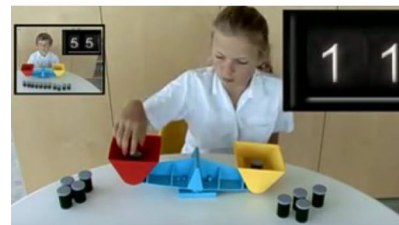
- 歩行者用信号
 - 身近でよい事例
- ブロックで作成は可能
 - 3重の入れ子構造
 - for+if+for 理解可能？
- 小4でこの内容を授業
 - 中学・高校のプログラミングは???



本当は情報科学をやるべき(CSアンプラグド)

コンピュータの仕組みを教具で学ぼう

- 本: 「アンプラグドコンピュータサイエンス」(イーテキスト研究所)
- サイト: **CSアンプラグド** <http://csunplugged.jp> 本の内容を無償公開!
- 絵本: 「テラと7人の賢者」(学研)小1-3用絵本。ぜひ小学校の図書室へ!
- 動画: YouTubeで「csunplugged」で検索
天秤ソート https://www.youtube.com/watch?v=cVMKXKoGu_Y



(小学校のまとめ)プログラミングで学ぶこと

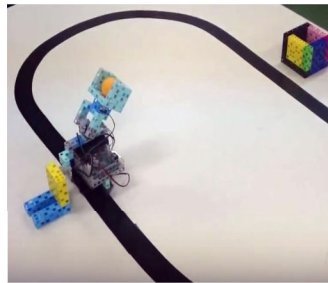
- プログラミング的思考
 - やりたいことを考え、コンピュータに言葉で伝える
- 論理的な考え方(の一部)
 - 究極の他人である機械に、簡潔に明瞭に伝える
 - 動かしながら形にしていく
- コンピュータの性質
 - コンピュータとともに生きていく
 - スマートフォン/アプリ: 人に合わせているが性質は機械
 - 時代が変わっても変わらない。セキュリティ、リスクを判断

小中高の連続したプログラミングの流れ(中学校)

- 小学校: 楽しさを体験する、嫌いにしない
 - 教科で手順(順次/反復/分岐)を活用
- 中学校: **世の仕組みを知る**
 - **計測・制御(あらゆる機器で利用)**
 - **双方向コンテンツ(アプリは通信)**
- 高校: 将来はセンター試験科目?
 - 情報I: アルゴリズム、シミュレーション
 - 情報II: データサイエンス、情報システムとプログラミング

計測制御と双方向コンテンツのプログラム例

- 計測・制御のプログラミング
 - 教材: ライトレース、時計のLED制御
 - ループでセンサ監視(if)+出力(on/off)
 - アナログ値、状態保持に発展可能?
- 双方向コンテンツのプログラミング
 - 事例: チャット(P2P)、Web???
 - 早急に技術面の整理が必要



ドリトル V3.2 (1 Dec 2017)

送信 受信

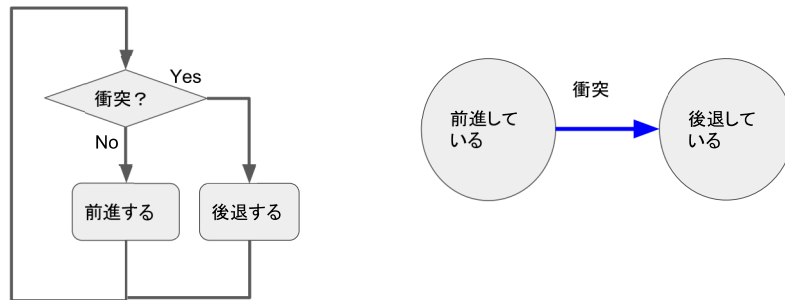
こんにちは
かめた: やっほー

```

プログラム [6/1]
1 送信ボタン=ボタン!"送信"作る。
2 送信枠=フィールド!作る。
3 受信ボタン=ボタン!"受信"作る 次の行。
4 受信枠=リスト!作る。
5 送信枠:動作=「
    
```

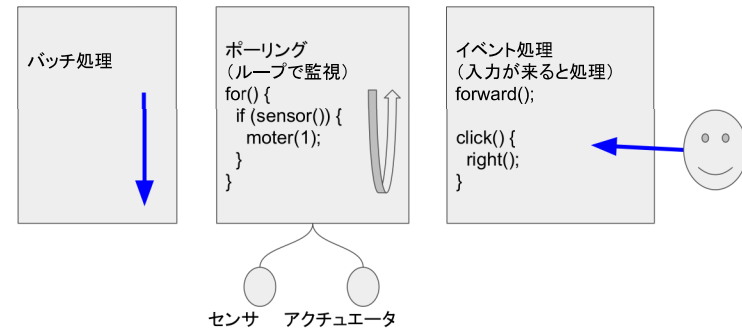
(補足2) 状態の図示(1)

- 手順と状態は両方とも必要
- 「壁に衝突したら後退する」の例



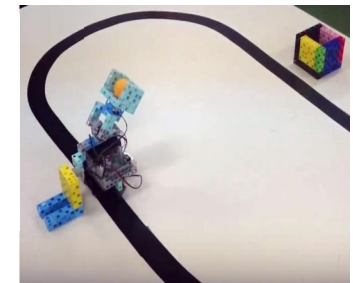
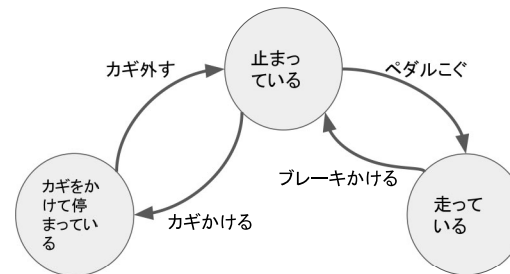
(補足1) 外部入力への処理

- 外部(人、デバイス、他のコンピュータ)とやり取り
- ポーリングとイベント処理が基本



(補足3) 状態の図示(2)

- フローチャート/アクティビティ図は動作(~する)
- 状態遷移図は状態(~している)
- 全体の状態を整理してから、個々の状態を手順で考える



小中高の連続したプログラミングの流れ(高校)

- 小学校: 楽しさを体験する、嫌いにしない
 - 教科で手順(順次/反復/分岐)を活用
- 中学校: 世の仕組みを知る
 - 計測・制御(あらゆる機器で利用)
 - 双方向コンテンツ(アプリは通信)
- 高校: 将来はセンター試験科目?
 - **情報I: アルゴリズム、シミュレーション**
 - **情報II: データサイエンス、情報システムとプログラミング**

アルゴリズムのプログラミング

- アルゴリズムはプログラミングの基本
 - 「知っている使わない」は構わないが知らないのは...
- 典型的なもの
 - **アルゴリズム**: 探索、整列、...
 - **処理**: 分割、再帰、...
 - **データ構造**: 配列、レコード、抽象データ型(スタック、キュー、集合、マップ、リストなど)、木構造
- 性能
 - 処理時間のオーダー
- 何をどこまで知っているべきか???

情報I, IIのプログラミング

- プログラミングは活用するために学ぶ
- 高校では何を学んでどこで使う?
- **小中は手順が中心**
 - 情報I, IIのプログラミングで**高校はデータを意識したい**

情報I	
(3)コンピュータとプログラミング	アルゴリズム、シミュレーション
(4)情報通信ネットワークとデータの利用	データ操作
情報II	
(3)情報とデータサイエンス	統計処理、グラフ等の視覚化
(4)情報システムとプログラミング	サーバーにデータを蓄積して利用

提案: 配列に代わる集合データ型の利点

- 情報Iのプログラミングでデータを扱おう
- 複数のデータと言えば配列だが...
- 配列のインデックスを省略できる言語を使いたい

配列aの要素を表示する例

```
int i;
for (i=0; i<5; i++) {
    print a[i];
}

foreach d in a {
    print d;
}
```

a

30	50	80	70	40
----	----	----	----	----

a[0] a[1] a[2] a[3] a[4]

a

30	50	80	70	40
----	----	----	----	----

探索アルゴリズムの例

- 整列は難しいが、探索なら扱いやすい

値が3以下の要素を配列bに格納する例

<pre>int i, j=0; for (i=0; i<5; i++) { if (a[i]<=3) { b[j]=a[i]; j++; } }</pre>	<pre>foreach d in a { if (d<=3) { b.add(d); } }</pre>
---	--

数だけを扱っても嬉しくない！

- 1次元配列だと1種類の値のみ
- 2次元配列だと全体が同じ型！orz
- 構造体の配列は生徒が理解不能...

配列	2次元配列	データフレーム																		
<pre>[0] 30 [1] 50 [2] 80 [3] 70 ...</pre>	<pre>[0] [1] [0] 30 50 [1] 50 70 [2] 80 75 [3] 70 80</pre>	<table border="1"> <thead> <tr> <th>名前</th> <th>数学</th> <th>英語</th> </tr> </thead> <tbody> <tr> <td>田中</td> <td>30</td> <td>50</td> </tr> <tr> <td>能城</td> <td>50</td> <td>70</td> </tr> <tr> <td>鎌田</td> <td>80</td> <td>75</td> </tr> <tr> <td>小松</td> <td>70</td> <td>80</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> </tbody> </table>	名前	数学	英語	田中	30	50	能城	50	70	鎌田	80	75	小松	70	80
名前	数学	英語																		
田中	30	50																		
能城	50	70																		
鎌田	80	75																		
小松	70	80																		
...																		

データベース的なデータ操作

- 言語にデータ操作を入れればSQLが不要に
- 大量のデータから扱いたいデータを取り出す
- データは表形式。2次元配列を簡単に扱いたい

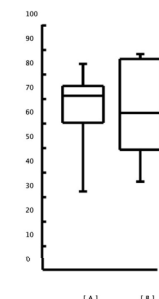
	プログラミング言語での利用	学習ツール(サクセス)
選択	<code>tb = ta.selection(price<3);</code>	選択 時間帯 夕方
結合	<code>tb = ta.join(tc);</code>	結合 商品データ

データ操作と統計処理(ドリトル)

- データ操作と統計処理の命令を提供
- グラフィックス等から一貫した学習

システム！"db" 使う。
 商品=テーブル！"商品.txt" ファイルから作る。
 売上=テーブル！"売上.txt" ファイルから作る。
 結果1=商品！（売上）結合。
 結果1！表示。
 結果2=結果1！"年齢" 度数。
 結果2！"度数" "年齢" 折れ線グラフ。
 結果2！表示。

```
[番号組国数理社英]
[60 A 74 48 87 71 28]
[59 A 37 56 67 56 85]
[55 A 60 60 67 38 65]
[54 A 74 62 87 31 28]
[51 A 38 50 53 75]
[49 A 74 49 92 74 28]
[48 A 75 57 54 66 85]
[44 A 60 53 67 28 65]
[43 A 74 48 65 44 28]
[42 A 75 57 43 54 85]
[38 A 60 55 84 34 65]
[37 A 74 54 38 54 28]
[36 A 75 58 23 54 85]
[32 A 60 62 64 51 65]
[31 A 74 52 31 51 28]
[26 A 39 50 65 64 74]
[25 A 49 48 75 54 57]
[24 A 68 56 37 75 38]
[22 A 59 65 58 61 48]
```

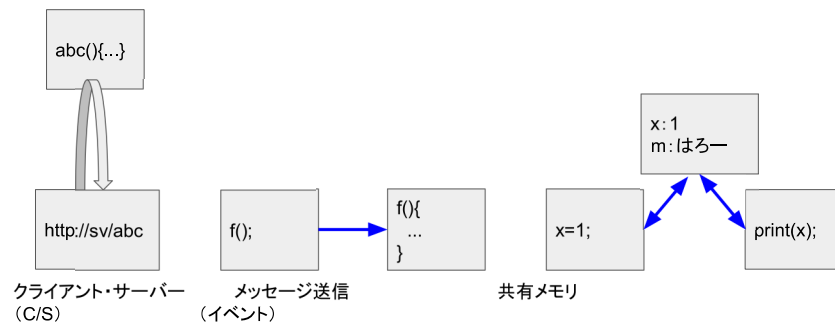


情報システムとプログラミング

- 情報システムは見えない縁の下の力持ち。
- 見えないものは「体験」で理解できる。
- 情報システムもプログラミングで体験的に理解したい。
- 必要な体験
 - サーバーによるサービスの提供
 - データベースによるデータ管理
 - クライアントからのアクセス
- はじめての試み: 実践例が必要

(補足4) 双方向通信のモデル

- クライアント・サーバー (Web)
- メッセージ送信 (Scratch)
- 共有メモリ (ドリトル)



IoT的なプログラミング学習の例(ドリトル)

- ラズベリーパイ+センサで計測 → サーバーに送信
- サーバーでデータを蓄積
- クライアントからデータにアクセスして利用

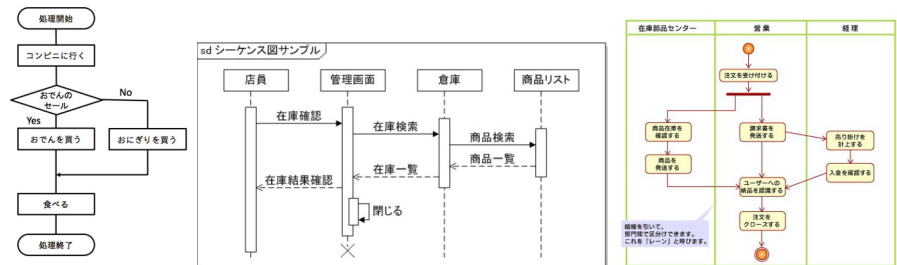
// データ受信のサンプルイメージ

サーバー! "172.16.86.21" 接続。
かめた=タートル! 作る。
かめた! 400 歩く 400 戻る 90 左
回り 200 歩く 200 戻る 図形を作
る。
タイマー! 作る 2 間隔「|n|
値=サーバー! "data" 読む。
かめた! (n*10)<値*5>位置。
」実行。

The graph on the right shows a line plot with '歩数' (Steps) on the x-axis (0 to 50) and '値' (Value) on the y-axis (0 to 200). The plot shows a series of peaks and troughs, indicating data being received and processed over time.

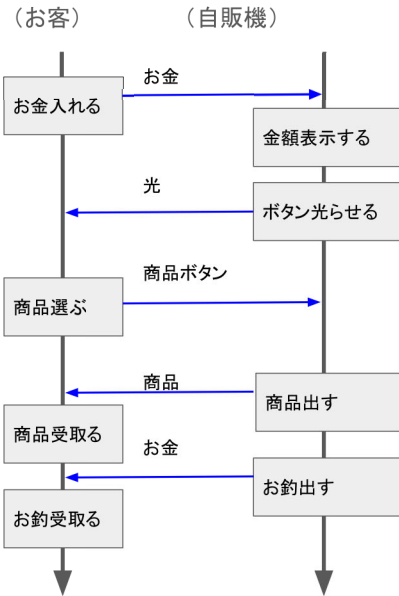
(補足5) 手順の図示

- フローチャート(手順) 処理は書けるがやり取りは書けない
- シーケンス図(通信) やり取りは書けるが処理は書かない
- アクティビティ図(通信と手順) やり取りと簡単な処理を書ける



提案

- 中学校段階では「**分岐や反復のないアクティビティ図**」を推奨
(**教育用アクティビティ図**)
 - 手順は順次のみで簡潔に記述
 - やり取りする情報に注目する
- 抽象度を意識した設計
 - システム全体の設計は**教育用アクティビティ図**や**状態遷移図(概要設計)**
 - 個々の処理(矩形)や状態(円)の実装は**フローチャート(詳細設計)**
- (自販機の例)
 - 「金額で買える商品だけボタンを光らせる」
 - 「お釣があればお釣を出す」という条件分岐は実装のため詳細設計で検討



よかったら使ってみてください(言語、教材)

- 教育用言語「**ドリトル**」
 - 小学校:算数(正多角形)など
 - 中学校:Artecブロック、micro:bit、双方向コンテンツ
 - 高校:シミュレーション、データ操作、統計処理
- 教育用オンラインプログラミング環境「**Bit Arrow**」
 - ドリトル、教育用JavaScript、C、DNCL
- データベースオンライン学習環境「**サクセス**」
- 情報科学学習教材「**CSアンプラグド**」

よかったら読んでみてください(書籍)



(いまやっていること)DNCL処理系「どんくり」

- 入試・資格試験問題用の仮想プログラミング言語
 - 大学入試センター試験、基本情報技術者試験など
 - DNCL:センター試験用。処理系(PEN)も存在
- アルゴリズム記述を想定 → 情報Iでも使えるのでは？
- オンラインの処理系「どんくり」を作ってみた

まとめ

- 今日のテーマは高等学校情報科の情報I, II
 - 直近には白紙からの教育を念頭に置きつつ、
 - 将来的には小中学校のプログラミングを踏まえる
- 小学校は体験と楽しさ。もっと学びたい！ → 中高へ
- 中学校はセンサ/アクチュエータや他のPCとやり取り
- 高校は情報Iで基礎を体験しつつ、情報IIでより深く
 - 手順からデータ操作へ
 - 教育用の言語、ライブラリの必要性
 - シミュレーション、データベース、データサイエンスでも
- カメやネコのいないプログラミングの世界へ！

「どんくり」

- 日本語表記は「読みやすい」が「書きにくい」
- Cのように書いて、DNCLで見られたら便利？相互変換を可能にしてみた

DNCL	↔	DNCLのC言語風表示
<pre>Values← {29,7,48,39,19,37,25,33,27,34,19,35,12, 5,14,30,47,3,10,32,6,36,25,4,7,38,1,18, 50,28,12} target←50 length←要素数(Values) i←1 i≤lengthかつValues[i]≠targetの間、 iを1増やす を繰り返す もしi≤lengthならば iと「番目に発見しました」を表示する を実行し、そうでなければ 「見つかりませんでした」を表示する を実行する</pre>		<pre>Values={29,7,48,39,19,37,25,33,27,34,19, 35,12,5,14,30,47,3,10,32,6,36,25,4,7,3 8,1,18,50,28,12}; target=50; length=count(Values); i=1; while(i<=length&&Values[i]!=target){ i+=1; } if(i<=length){ print(i,"番目に発見しました"); }else{ print("見つかりませんでした"); }</pre>