第１章：プログラミングって何だろう？

これから「プログラミングとは何か？」を説明していくよ。

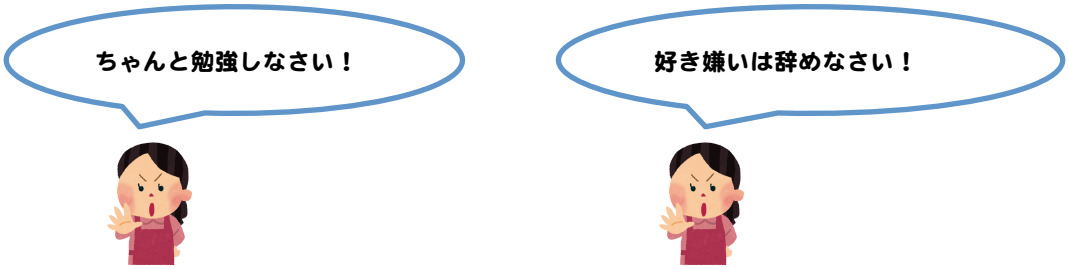
みんなは「プログラミング」と聞いてどんな事を思い浮かべるかな？

・ポケモン Go とかパズドラとかのスマホゲームを作れるもの？
・LINE みたいなスマートフォンで使うアプリを作れるもの？
・２ch とか Youtube とか誰でも知っているような web サイトを作れるもの？

実はこれ、全部プログラミングで作ることができるんだ。

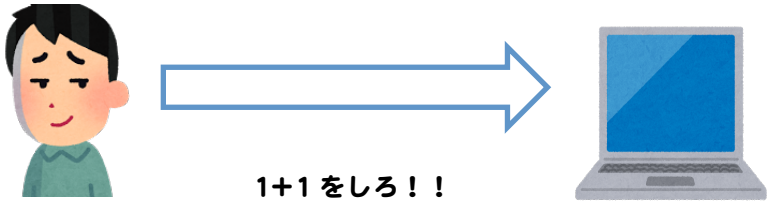
プログラミングはパソコン上でできる命令の集まりのようなもので
上に挙げたようなものは様々な命令が組み合わさって出来ているんだ。

みんなのなかでの命令って言うと・・・



みたいなことを思い浮かべるよね。

プログラミングでは、逆にみんながパソコンに対して命令することができるんだ。

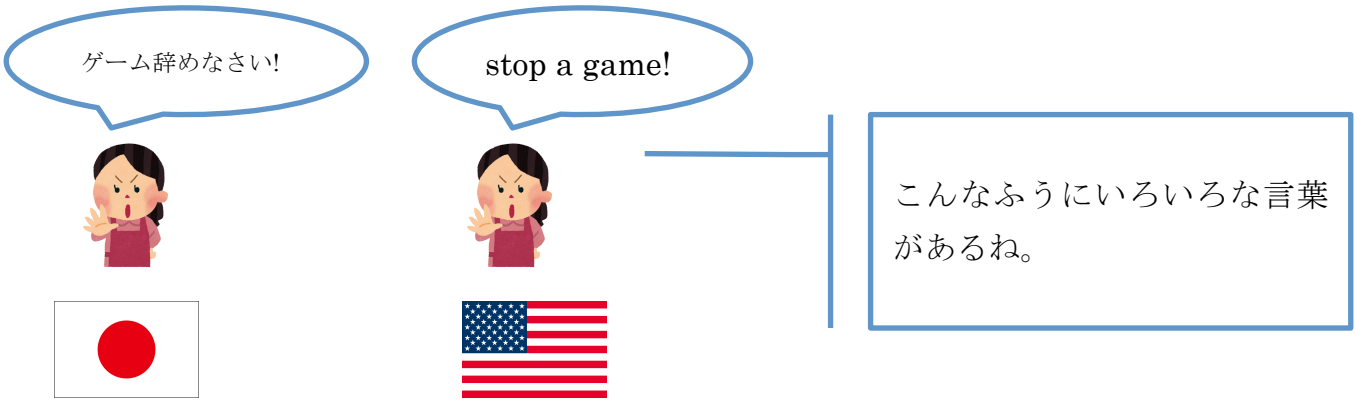


プログラミングをするということは、
こんなふうにパソコンに対しての命令を書く事をいうよ。

この命令をたくさん組み合わせる事で、それがかみあって
Line や Youtube のような、みんなが思い浮かべるような１つのシステムが出来上がるのだ。

ところで、今私たちが使っている言葉には、様々な言語があるよね。
英語・ドイツ語・日本語・中国語・韓国語などなど・・・
私たちは、こういった言語を場所だったり用途だったりによって使い分けているよね。

例えば、アメリカに行けば英語を使うだろうし、韓国に行けば韓国語を使うだろう。
大勢の国の人たちが集まる会議では、みんな英語を使うかもしれないよね。



これと同じようにプログラミングにも“言語”というものがあって
用途や目的によって言語を使い分けているんだ。

例えば、
windows のパソコンソフトを作りたい場合は、
VisualBasic という言語や C# という言語を使うことが多いけど、
２ch のようなインターネット上の掲示板は、
PHP という言語や Ruby という言語で作られる事が多いんだ。

こんなふうに作りたいものに合った言語を選んでプログラミングをしていこう！

実際にプログラミングをしていくときには

“コード”というパソコンに対する命令を書いていくよ。

例えば、みんながコンビニに行って２５０円のりんごとみかんを１つずつ

買い物をして、１０００円払ったときのおつりを計算するとき、

下の例のような流れになるね。

例）

a 買う商品の合計金額を計算

b 出したお金の合計を入力

↓

b-a をしておつりを計算


＋


２５０円　　２５０円

(a) ＝５００円！！

この場合、

(b) 

```
a=500
b=1000
prin (b-a)
```

というコードを書くと上の例の計算を実行できるんだ。

今はこのコードの意味が分からなくても問題ないよ。

コードはパソコンへの命令だから、

このコードを読み込んだパソコンは命令を聞いておつりの計算をして

その結果を画面に表示してくれるということになるんだ

→ 500 円

今回は短くて簡単な処理の例だったけど、どんなプログラムでもこのコードと呼ばれているもので作られているのは変わりないんだ。

第2章:「文字列」ってなんだろう？

1. 文字とは？

突然ですがクイズです！

このひらがなたちをなんていう？

そう！「文字」っていうよね。

じゃあ、英語はどうだろう？

そう！英語も「文字」っていうよね！

つまり「文字」というのは

私たちがいつもノートに書いてる言葉のことをいいます。

☐練習してみよう！

次のうち「文字」であるものはどれか。記号で答えなさい。

㋐か　㋑B　㋒ほ　㋓〜　㋔G　　答え：

ア,イ,ウ,オ

２．列とは？

みんなありって知ってる？

ありってよく行列を作って歩いてるよね。

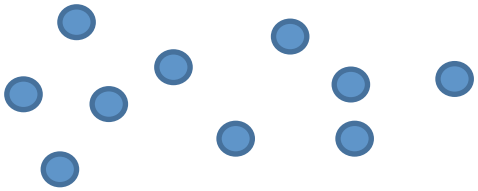
私たちも一緒で、こんでいるレストランに入ろうとしたら、列を作って待っているよね。

つまり「列」というのは、何かが並んでいることをいいます！

□練習してみよう！

次のうち□が「列」になっていないものはどれか。記号で答えなさい。

㋐ 

㋑ 

答え：　イ

３．文字列とは？

みんな、「文字」「列」の意味は分かったかな？

では今回のテーマである「文字列」とは何だろう？

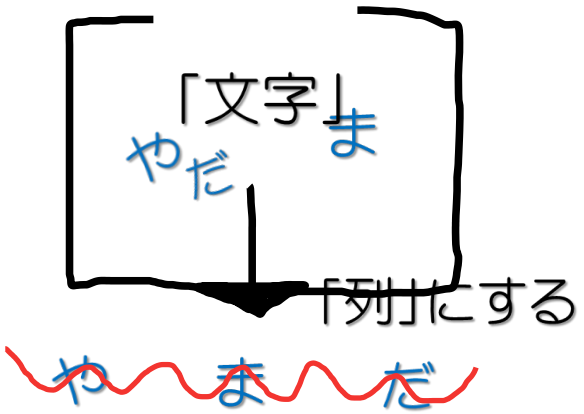
これまでの知識を使って考えてみよう！

「文字列」とは、言葉の通り文字が列になったものです。

今までの知識を使うと…

- 「文字」…いつもノートに書いている言葉

- 「列」…何かが並んでいること



つまり！文字がんでいることを「文字列」といいます！

☐練習してみよう！

下のバラバラのひらがなを、くだものの名前の文字列にならべ直しなさい

(1) ご　ん　り

答え：　りんご

(2) さ　ら　く　ん　ぼ

答え：　さくらんぼ

「Print文」ってなんだろう？

～ほかのブラウザで《文字列》を表示してみよう！～

今回学んだ《文字列》をためしにほかのブラウザで表示してみよう！
他のブラウザに文字列を表示するには「Print 文」というものを使ってコードを書いていきます。

例えば…？

２つのＡというパソコンとＢという画面があるとします。
ちなみにＡのパソコンで書いたコードはＢに表示されることとします。

今Ｂの画面には何も書いてありません。そこで、Ｂのパソコンに「こんにちは」と表示させたいとします。
でも、Ｂにはキーボードがないので、表示させることができません。
じゃあ、どうやって「こんにちは」って表示するの！？
それはＡのパソコンを利用することで簡単に解決できちゃいます！

パソコンにコードを書こう！
Ａのパソコンからの画面に「文字列」を表示させるために実は**魔法の言葉**があるのです！
その言葉とは「Print」という言葉です。
「Print」という言葉の後に「こんにちは」と書くことでＢの画面になんと
「こんにちは」を表示させることができちゃうのです！
では、実際にコードを書いてみましょう。
…と、ちょっとその前に…

コードを書く時のおまじないを覚えよう！

コードを書くとき、文字列を記入するときは

「u(”文字列”)」という風に書こう！

このおまじないを使って書いてみると…

```
Print u(“こんにちは”)
```

簡単！１行書くだけでＡのパソコンからＢの画面に「こんにちは」を表示させることができちゃうのですよ！

練習してみよう！

Print 文を用いて次の文字列を画面に表示させるためのコードを書きなさい。

(1)イチゴ

答え：　Print u(“イチゴ”)

(2)マジ LOVE1000%

答え：　Print u(“マジ LOVE1000%”)

(3)抹茶っておいしいよね！

答え：　Print u(“抹茶っておいしいよね！”)

第３章　数値計算

　みなさんは「１＋１＝２」、「５−３＝２」のような計算をしたことがあると思います。
　プログラミングではこれと似たように数値を計算するができます。

まず、はじめに覚えておいて欲しいのは「書き方」です。

簡単な計算の書き方

Python で簡単な計算をするときには
Print(〇　△　〇)
〇に数字を入れて、
△には計算するときに使う記号を入れます。
例:print(10 + 5)

次に計算するときにはちゃんとしたルールがあります。

計算するときのルール

ルール１：整数と少数の演算　→　少数
例:
```
print(5 + 2.0)
```
答え：7.0

ルール２：整数同士の割り算　→　切り捨ての整数
例:
```
print(10 / 3)
```
答え：3

ルール３：整数と少数の組み合わせの割り算で割り切れないとき　→　少数
例:
```
print(10 / 3.0)
```
答え 3.33333

例に書いてあることはよくわからないかもしれませんが、この後説明します。

普段、みなさんは計算式を書く時には+や-、×、÷を使いますよね？
プログラミングではちょっと違う記号を使って計算をします。
どんなものがあるのかを表でまとめてみました！

記号とその意味

記号	+	-	*	/	//	%	**
意味	足し算	引き算	掛け算	割り算	切り捨て割り算	余り	べき乗

これらの記号をさっきの△に入れて使うことができます！

次に記号の詳しい説明と使い方を説明します。

みなさんは足し算、引き算、掛け算、割り算は知っていると思います。
Python ではそれぞれ次のように書きます。
例:

```
print(4 + 5)
print(10 - 3)
print(6 * 3)
print(8 / 4)
```

答えは上から順に 9、7、18、2 となりますね！
でも割り算ってちゃんと割り切れないことがありますよね。
そんな時はさっきのルール２を確認してみよう！

余りとは...
割り算の時に割り切れないことがありますよね？
9÷2＝4...1
↑これ！
例:

```
print(13 % 4)
```

答えは 1 ですね！

切り捨て割り算とは...
みなさんは少数の割り算をしたことがありますよね？
例えば、　99.99÷3＝33.33　ですよね？
切り捨て割り算では、答えで出た数の小数点以下の数がどんな数でも 0 にします。
つまり、99.99÷3=33.0 という答えになります。
例:

```
print(88.22 // 4)
```

答えは 22.0 ですね！

べき乗はおそらく中学一年生で習うと思います。
普通に手で書く時は右のように書きます。
大きい数字を小さい数字の数だけかけていくことを表しています。
つまり右の場合だと、2×2×2＝8　ですね！

$$2^3$$

例:

```
print(3 ** 3)
```

答えは 27 ですね！

自分で計算してみよう

問題：次の計算式を入力して答えを出してください。() で指示されている場合は指示された通りに数値を出してください。

1. 1 + 7
2. 24 - 17.0
3. 5.6 × 3.5
4. 192 ÷ 4.0
5. 35 ÷ 6　　(余り)
6. 60.03 ÷ 3　　(小数点以下切り捨て)
7. 4 × 4 × 4 × 4　　(べき乗で計算)

解答

1. 8
2. 7.0
3. 19.6
4. 48.0
5. 5
6. 20.0
7. 256

「変数」ってなんだろう？

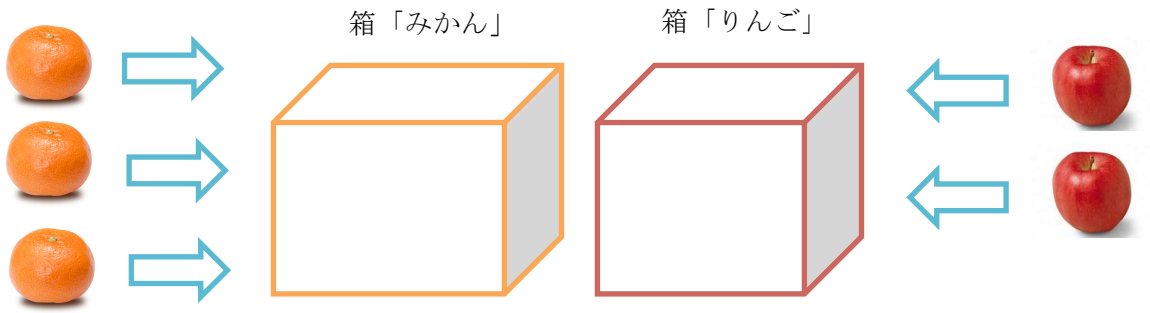
・まずはイメージをつかんでみよう！

例：

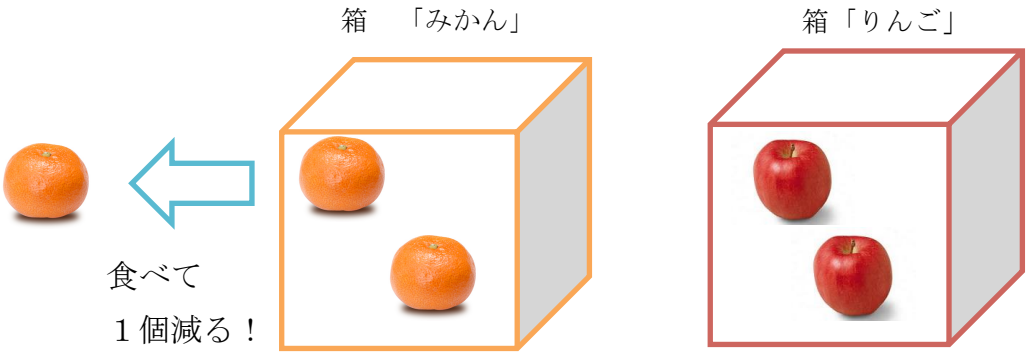
太郎君はみかんとりんごを友達からもらったので、箱を2つ用意しました。

もらったみかんとりんごを全部手に持つのは大変なので、
箱に一旦入れて数を数える事にしました。

「みかん」という箱にみかんを3個入れ、
「りんご」という箱にりんごを2個入れました。



しかし、太郎君はみかんを1つ食べてしまいました。



みかんの箱のみかんとりんごの箱のりんごを全部あわせると
今現在、合計何個でしょうか？

答え：4個

最初はみかんの箱には3個、りんごの箱には2個入っているね。

プログラミングの世界ではこの箱のことを「変数」と言うよ。

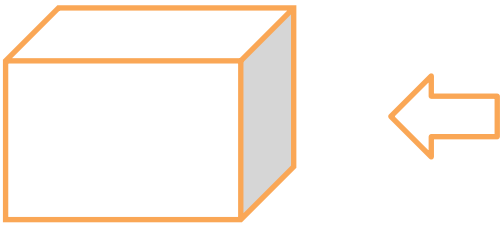
もらったみかんとりんごをずっと手に持っていては
他の事が出来なくて不便なので一旦箱にしまうよね。
プログラミングでも同じように変数にものをしまうんだ。

現実には、みかんそのものを箱に入れているけど
プログラミングはパソコン上で行うから
みかんを箱に入れる事はできないんだ。

だから、「みかんは〇〇個あるよ」という”数字”を変数に入れるんだ。

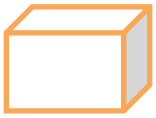
みかんの箱にみかんを3つ入れる式はプログラングでは
みかん＝3
となるんだ。

変数「みかん」

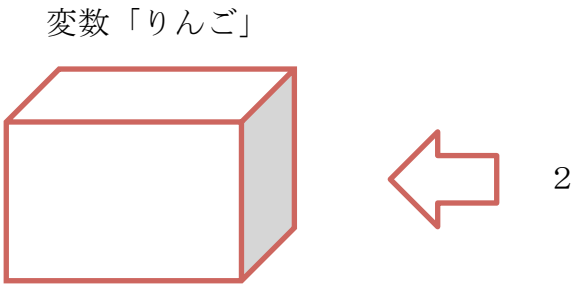


ここでの「＝」は「同じだよ」という意味ではなく、
“変数「みかん」に「3」という数字を入れるよ“という意味になるよ。

これで変数「みかん」の中身は「3」だから、

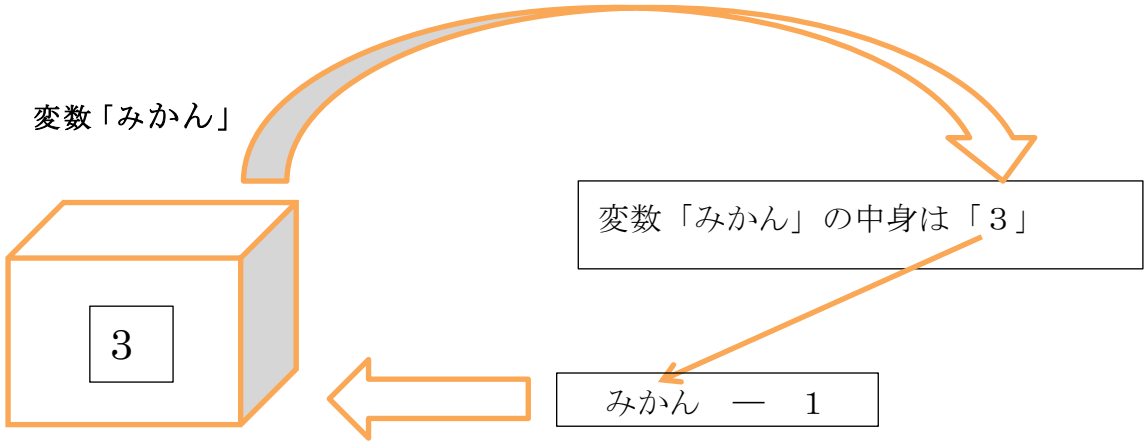
「3」 と 変数「みかん」  は同じものとして考えることが出来るんだよ。

同じように、
りんご=2
とすると、変数りんごに「2」を入れられるよ。



みかんを食べると箱からみかんが減るように、変数の中身も変わっていくんだ。

さっきみたいに、みかんの箱からみかんが1個減ったときは、
プログラミングでは、
みかん=みかん−1
という式であらわせるよ。



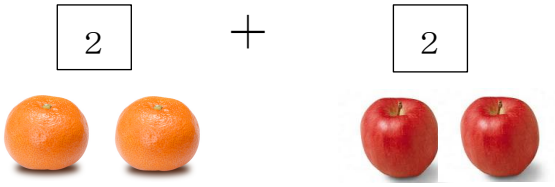
変数「みかん」に計算
結果を入れているよ。

これをすると、”変数「みかん」−1”すなわち”3－1”の結果の
“2”が新しい変数「みかん」の中身になるね。

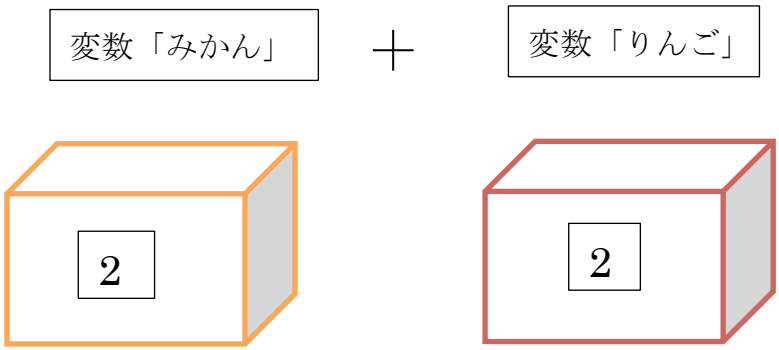
最後に箱の中身を足し算してみよう！

みんなが実際に数を数える時は、
“（みかんの箱の）2個＋（リンゴの箱の）2個”
というように数字で計算するよね。

でも、プログラミングでは、
“変数「みかん」＋変数「りんご」”というように
変数同士を足し算しても中身の数字を計算してくれて同じ結果になるよ。



をプログラミングで表すと・・・



と表せるんだ。

・練習問題：実際に自分で書いてみよう！

問題：変数 mikan に3、変数 ringo に2を代入して、mikan と ringo を足し算してその結果を出力してください。

解答

```
mikan=3
ringo=2
print（mikan + ringo）
```

・「条件分岐(じょうけんぶんき)」ってなんだろう?

・まずはイメージをつかんでみよう!

例: みんなはお母さんにこんなことを言われたことはないかい?

その1.「テストで100点を取ったら、ケーキを買ってあげる。でも100点を取れなかったら、 何も買ってあげない!」

その2.「次の野球の試合でホームランを打ったら1000円お小遣いをあげる。ヒットを打ったら 500 円あげる。でもダメダメだったら 0 円ね。」

これは、実は二つとも条件分岐の一つなんだ。

一つ目から詳しく見てみよう!

(もしも)テストで100点を取ったら、ケーキを買ってあげる。 **でも(もしも)100点をとれなかったら**、何も買ってあげない!

次に二つ目を見てみよう!

(もしも)次の野球の試合でホームランを打ったら、1000 円お小遣いをあげる。 **(それか、もしも)ヒットを打ったら**、500 円あげる。 **でも(もしも)ダメダメだったら**、0 円ね。

なんとなくパターンがあるのがわかったかな?

上の 2 つの文は、どちらも条件とその時の結果を表しているんだ! **もしも、こんな感じだったら(条件)**、こうしよう!(結果) という感じだね!

例題:条件文を書いてみよう! 解答例1:お腹が空いたら、ご飯を食べよう。でも、お腹がいっぱいになったら、食べるのをや めよう。解答例2:部屋が汚かったら、嫌な気分。でも、部屋がキレイだったら、いい気分!

***

それじゃあ、今まで勉強した変数の考え方を使って上の条件分岐の考え方について書いてみる よ。

まず1つ目の文について考えてみよう!テストの点数を score という変数を使って表してみることにするよ。 None は英語で、何もなし、という意味だよ。ここでは、君は 100 点を取れたことにしよう!

score = 100

if score == 100:

print 'Cake!!!'

else:

次に 2 つ目の文について考えてみよう!野球の結果を result という変数を使って表してみることにするよ。 ここでは、君はホームランを打ったとしよう!

result = ‘homerun’

if result == 'homerun':

print '1000yen!!!'

elif result == "hit":

print '500yen!!'

---

print 'None...'

---

else:

プログラミングっぽくなってきたね! さて、ここで新しい記号や言葉が出てきたよ。一つ一つ説明していくね。

ここで、 if は日本語でいう「もしも」にあたるよ。Elif は、 else と if をあわせたもので、「それか、もしも」にあたるよ。 else は「でも、もしも」にあたるよ。

この 3 つの言葉(おまじないのようなものだよ)の後に条件を書いて、その次の行にその時の 結果を書くんだよ。上の例で言うと、**太文字のとこ**

ろが条件、波線のところが結果を書いている のがわかるかな?下の日本語とどう対応しているか、確認してみてね。

>まずは一つ目から見てみよう!>**>(もしも)テストで100点を取ったら**、ケーキを買ってあげる。 **>でも(もしも)100点をとれなかったら**、何も買ってあげない!>>次に二つ目を見てみよう!>**>(もしも)次の野球の試合でホームランを打ったら**、1000 円お小遣いをあげる。 **>(それか、もしも)ヒットを打ったら**、500 円あげる。 **>でも(もしも)ダメダメだったら**、0 円ね。

さて、条件文の中にも見たことがない記号があったはずだ。 「==」という記号がそれにあたるね。この記号にはいくつか仲間が存在する。この仲間をまと

print '0yen ...'

めて「比較演算子」と呼ぶよ。一つ一つ見ていこう。

== a==b!= a != b>= a >= b> a > b<= a <= b< a < b ここで覚えるのは難しいと思うので、実際に使ってみて覚えてほしい。

bがaに等しいb が a に等しくない bよりaが大きいか等しい b より a が大きい bよりaが小さいか等しい b より a が小さい

それから、もし条件文に二つ以上の条件を入れたい時は、その条件を and でつなごう! 例:if score >= 50 and score <= 70

じゃあ、実際に例題を出すので解いてみてね!

例題:もしも、今日のごはんがおいしかったら、 happy!! と表示しよう。でも、もしも今日のご飯が失敗だったら(おいしくなかったら)、unhappy...と表示しよう。 ごはんを、変数 gohan 、条件のおいしいを oishi を使って書いてみてね。

答え:if gohan == "oishi":

print "huppy!!" else:

print "unhuppy..." ***

```
if gohan == "oishi": print "huppy!!"

elif gohan != "oishi": print "unhuppy..."
```

例題:もしも、テストの結果が 80 点以上だったら、 perfect! と表示しよう。 それか、もしもテストの結果が 60 点以上 80 点未満だったら、soso... と表示しよう。 でも、もしもテストの結果が 60 点未満だったら、 umm...と表示しよう。 テストの結果を、変数 test を使って書いてみてね。

答え:

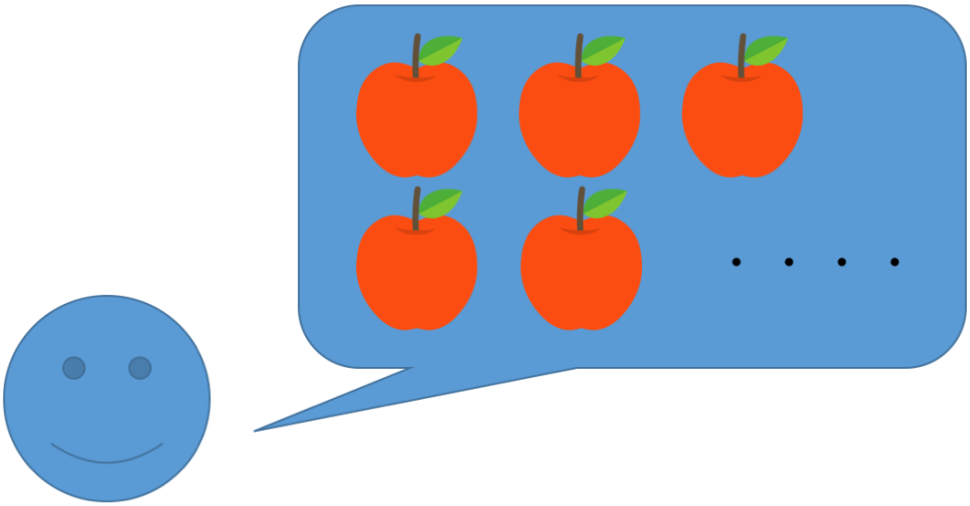
```
if test >= 80:print "perfect!!!"

elif test >= 60 and test < 80: print "soso..."

else:print "umm..."
```

くりかえし処理

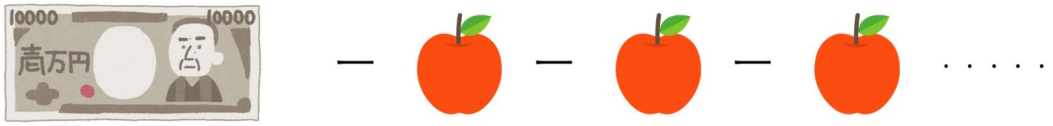
プログラムを書いていると同じ計算をたくさんしなくてはいけない時がある。
どんな時だろう？毎日の生活を例に考えてみよう。

問題

太郎くんはリンゴが大好きなので持っているお金で買えるだけリンゴを買うことにしました。
リンゴは何個まで買うことができるでしょうか？



さてどんなプログラムを書いたらキレイに計算ができるんだろう。
今回はリンゴの値段と、太郎くんが持ってるお金の２つの変数が出てきそうだね。
決まった数のリンゴを買うのは「数値の計算」でやったね！
でも今回は買えるだけリンゴを買わないといけないんだ
お金が足りなくなるまでリンゴを買うんだから何回も引き算をしていけばいいね。



でも何回引き算をしたらいいのかわからないんだ。
だから私たちはこの計算を考える時、

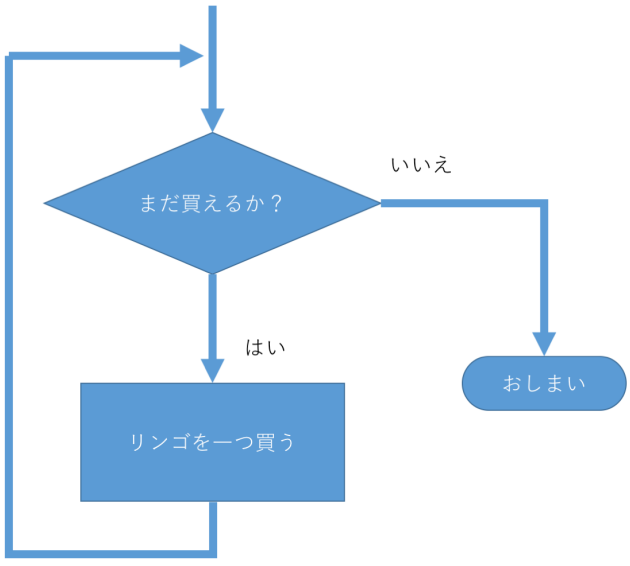
① 残りのお金を見る。
もし、まだお金があまっていれば
もう一個リンゴを買う。

② 残りのお金を見る。
もし、まだお金があまっていれば
もう一個リンゴを買う。

③ 残りのお金を見る。
もし、まだお金があまっていれば
もう一個リンゴを買う。

こんな風に何回も同じことをくりかえして考えているんだ。

これをコンピュータにやらせるのがくりかえし処理だよ。
人間と違ってコンピュータは同じことを何度も早く、くりかえすのが得意なんだ。



人間なら考える時間がひつような問題もコンピュータならすぐだよ。
同じことを何度もやらせるのに便利なのが「条件分岐」なんだ。

- お金が十分残っていればもう一つ買う。
- もし足りなければおしまい。

こんな条件分岐を使えばいいね。

このグルグルと回っているのがくりかえし処理の正体だ。
ループと呼ぶこともあるよ。
今回は、「条件分岐」を使うことで回数のわからないくりかえし処理に挑戦したけど、リンゴを１０個買うなど個数が決まっていればもっと簡単に書くことができるよ。

練習問題

- 自動販売機に 1000 円を入れて 150 円のジュースを５本買いました。お釣りは何円でしょう。

- 花子さんは 3000 円を持って、リンゴが一個 100 円、ミカンが 200 円の八百屋さんでリンゴ２つ買った後にミカンを買えるだけ買いました。さて何円残ったでしょうか。条件分岐をつかった、くりかえし処理で計算してみよう。

関数ってなんだろう？

・まずはイメージをつかんでみよう！

120 円を入れ自動販売機でジュースを買うことにしよう！るとジュースが 1 本買えるね。
このように「ある値を入れて、違う値をただ一つ返すこと」を関数というんだ！

ちょっとわかりづらいかもしれないから、例題を出してみるよ！

・例題 1
100問(1問1点)の漢字テストをやったとき
A くんは 80 点、B くんは 45 点だったとします。それぞれが間違えた問題の数は？

答え A くん 20 問、B くん 55 問

解説
答えはすぐに出たよね。でも、ここで大事なのは答えを出すことではないんだ。
A くんに注目するよ。20 問という答えは 80 点の時にしか出ないよね。」
そう、「ある値を入れて、違う値をただ一つ返すこと」にあっているよね。
だから例題 1 は関数と呼べるんだ！

例題 2
身長 150cm の C くんの体重は？

答え たくさん

解説
答えはたくさんでたね。身長 150cm という 1 つの値を入れて、違う値がたくさん出てしまったから、例題 2 は関数とは言わないんだ。

だいたい関数のことはわかったかな？

じゃあ実際にプログラミングで関数をやってみよう！

Python で関数を使う時には、
関数の名前を決めてやらないといけないんだ！
ここでは、hello というなまえにするよ。

```
def hello():
```

このように、def の後ろに名前と()をつけて最後に ： をつけるんだ。
「:」これを忘れやすいからきをつけてね！

じゃあ次に、この関数になにをしてもらうのかを決めよう！
今回は、関数 hello に hello world と出力させることにしよう。
その時は、

```
def hello():
    print (“hello world”)
```

こうやって書くんだ。さっき名前を決めたその下にその関数にしてほしい命令を書くだけでいいんだ！

次に実際に出力してみよう！
その時のコードは

```
def hello():
    print(“hello  world”)
hello()
```

これで表示することができるんだ。
そう、hello()と書くだけでいいんだ。
いちいち「”hello　world”」と書かなくても、「hello()」と書くだけでいいんだ。とても楽になるよね。これが関数のいいところなんだ。
しかも、この命令は自分で好きなように決めていいんだ！

よし、じゃあ実際に練習問題を解いてみよう！

問 1
hello という名前で、「hello rikadai」と出力する関数を作ろう。

問 2
add　という名前で、x と y の合計を出す関数を作ろう。

問 1　答え

```
def hello():
    print(“hello rikadai”)
hello()
```

問 2　答え

```
def add(x,y):
    print(x+y)
add(X,Y)
```

→X、Y には好きな数字が入るよ！