



情報活用能力としてのねらい	活用できる教科や学習活動
<p><b>C6：データの傾向</b>                      ・表やグラフを用いてデータを統計的に処理することができる</p> <p><b>C4：問題解決の手順</b>                      ・問題解決の手続きを順序・繰り返し・分岐などを組み合わせて表現できる</p> <p><b>C5：試行錯誤</b>                      ・試作やシミュレーションを通して問題解決の方向性や改善策を見いだす</p>	<p>2年数学                      「確率～ことからの起こりやすさの観察～」</p>

## 2年 数学「確率 ～ことからの起こりやすさの観察～」

時間	学習活動	準備物・留意点等
5分	<p><b>1. 考えてみよう</b></p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">サイコロの「1が出る確率」を調べよう</div> <p>・「1が出る確率」の相対度数を求める。</p>	<p>【準備物】</p> <ul style="list-style-type: none"> <li>・サイコロ (可能であれば)</li> </ul> <p>・実際にサイコロを振って計算結果を確かめようとすると時間がかかることを確認する。</p>
12分	<p><b>2. 作ってみよう</b></p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">サイコロを振るプログラムをScratchで作ろう</div> <p>・ステップを分けながらScratchでプログラミングをする。</p>	<ul style="list-style-type: none"> <li>・数学の学習に重点を置きたい場合は、作成動画を確認させながら、プログラミングさせる。</li> <li>・プログラミングの学習に重点を置きたい場合は、それぞれのブロックの意味や、変数のもつ特性などを確認しながら進める。</li> </ul>
8分	<p><b>3. 動かしてみよう</b></p> <p>・作ったプログラムが正しく動くか確認する。</p>	<ul style="list-style-type: none"> <li>・一度でうまくいかないときは、結果を確かめながら、条件式を見直させる(デバッグ)。</li> </ul>
10分	<p><b>4. 調べよう</b></p> <p>・作ったプログラムを活用して、出た結果から手計算する。</p>	<ul style="list-style-type: none"> <li>・回数が増えるほど「1. 考えてみよう」で出した計算結果に近づいていく事に気づかせる。</li> </ul>
15分	<p><b>5. 改良しよう</b></p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">プログラムをさらに便利にしよう</div>	<ul style="list-style-type: none"> <li>・「繰り返し処理」を使い、多数回の試行結果をより効率よく得ることで、プログラムの良さに気づかせる。</li> <li>・数学の学習に重点を置きたい場合は、作成動画を確認させながらプログラミングさせ、試行回数が多くなればなるほど更に計算結果に近づくことを気づかせる。</li> </ul>



### 必要となるICT機器

・ **PC (Windows)**

- ※Scratch3.0を利用する場合は、ブラウザソフトGoogle Chrome , Mozilla Firefox , Microsoft Edgeを推奨
- ※IE (インターネットエクスプローラー) は非対応 (2019年2月現在)

・ **iPad (iOS)**

- ※Scratch3.0を利用する場合は、ブラウザソフトSafariを推奨
- ※アプリであれば、「Pyonkee」の利用を推奨



### 機器活用のポイント

- ・問題解決や表現活動を行う場合に、コンピュータに与える論理的手続きやデータを様々工夫できることを、ICTを活用したプログラミング言語の体験を通じて理解する。

・「1 が出る確率」を計算で出し、本当にそうなるのかをプログラミングで確かめることを伝える。

- ・ QRコードのリンク先の動画を見て、ステップを分けながら「1 が出た回数を調べるプログラム」を完成させる。
- ・ それぞれのブロックの持つ意味を確認する。
- ・ それぞれの「変数」がなぜその位置に来るのかを考えさせる。

・ 試行回数が増えるほど、最初の計算結果に近づいていくことを実感させる。

### 完成イメージ (例)

・ 解答は一例で、これ以外の解答もあることに留意する。

繰り返しの回数はサイコロを振りたい回数で変えることができる。

「変数を0にする」のブロックを演算の最初に入れ、すべての変数をリセットする。

プログラミング

#### サイコロの1が出る確率を調べるプログラム

**1.考えてみよう** サイコロの「1が出る確率」を調べよう  
サイコロを振ったとき「1が出る場合」の確率はいくらでしょう。計算で出した結果と、実際にサイコロを振った結果との程度の差が出るのでしょうか。

サイコロの目の出方  通り 「1」の目の出方  通り  
「1が出る確率」は、 $1 \div 6 = 0.166\dots$

**2.作ってみよう** 「1が出た回数を調べるプログラム」をScratchで作ろう  
[STEP1] サイコロを振るプログラムを作成。  
[STEP2] サイコロの1が出た「回数」を数えるプログラムに改良。

**STEP1** サイコロを振る

**このステップで使うブロック**

- ① プログラムをスタートさせるブロック
- ② 適合分けをするブロック
- ③ キャラクターにしゃべらせるブロック
- ④ 等式を判定するブロック
- ⑤ 特定の数をランダムに発生させるブロック

**① スキルのポイント①**  
STEP1のプログラムが完成したら、きちんと動くか確認してみましょう。  
緑の旗を押すとサイコロを振ります（実際には旗のキャラクターがサイコロの目をしゃべります）。

**STEP2** 1が出た「回数」を数える

**このステップで追加するブロック**

- ① 「変数」の数を足すブロック
- ② 「変数」の数を減らすブロック

**① スキルのポイント②**  
**変数の作り方**  
変数タブから「変数を作る」ボタンを押し、変数名（例：「1」が出た回数）を入力すると変数を作成します。

**3.動かしてみよう** プログラムを動かしてみよう

- 緑の旗を押すとサイコロを一回振ります。
- 「1が出た」場合は「1が出た」回数の変数が増えます。「1以外が出た」場合は「1以外が出た」回数の変数が増えます。
- 「1が出た」回数を「0」にするのブロックをクリックすると変数がすべて0になります。

**4.調べよう** プログラムの結果を記録しよう  
作ったプログラムを動かして、サイコロを振った回数と、1が出た回数を記録し、「1が出る確率」を計算しましょう。サイコロを振る回数が増えるとうなっていくでしょうか。

サイコロを振った回数	1が出た回数	1が出る確率
(例) 10回	2回	$2 \div 10 = 0.2$
10回	3回	$3 \div 10 = 0.3$
30回	7回	$7 \div 30 = 0.233$
100回	18回	$18 \div 100 = 0.18$

**5.改良しよう** プログラムをさらに便利しよう  
サイコロをたくさん振って、かつ、自動で「1が出る確率」を計算するプログラムにするにはどうしたら良いでしょうか？ 下のブロックを使ってプログラムを改良しましょう。

これらのブロックをヒントに考えよう

- こんにちは！と挨拶
- 15回繰り返す
- みんな！ハイ！
- 15回繰り返す

**このステップで追加するブロック**

- ① キャラクターにしゃべらせるブロック
- ② 文章をつなぐブロック
- ③ 繰り返すブロック
- ④ 特定の回数、演算を繰り返すブロック

**① スキルのポイント③**  
「繰り返しブロック」を使うと、ブロックの間に含まれた部分だけ、指定した回数だけ繰り返して処理されます。

サイコロを振った回数	1が出た回数	1が出る確率
500回	75回	$75 \div 500 = 0.15$
3000回	493回	$493 \div 3000 = 0.16433$

Scratch is a project of the Scratch Foundation, in collaboration with the Lifelong Kindergarten Group at the MIT Media Lab. It is available for free at <https://scratch.mit.edu>

### アドバイザーのコメント

プログラミングとしては、「変数を活用したプログラミング」が体験できます。ワークシートを行うことで【試行錯誤】や【問題解決の手順】の力を養うことができます。また、作成したプログラムの実行結果を表やグラフにまとめることで、【データの傾向】の力を養うこともできます。

数学の時間だけでなく、【技術・家庭科(技術分野)】の中の「生活や社会を支える情報の技術」で扱い、プログラミングの基礎的な知識及び技能を習得する学習や【小・中学校の総合的な学習の時間】などのプログラミングを学習する教材としての活用も考えられます。また、【小・中・高校教育課程内で各教科等とは別に実施するもの】の一例としての活用も可能です。