

レゴ エデュケーション® WeDo 2.0 プログラミング的思考

教師用ガイド



WeDo 2.0

目次

WeDo 2.0 プログラミング
的思考への導入

3-11

WeDo 2.0を使った学習
カリキュラム

12-28

WeDo 2.0を使った学習
評価

29-41



WeDo 2.0のプロジェクト学習で プログラミング的思考を身につける

この章では、科学的な設定の中でWeDo 2.0を使いながらプログラミング的思考スキルを身につける方法を説明します。





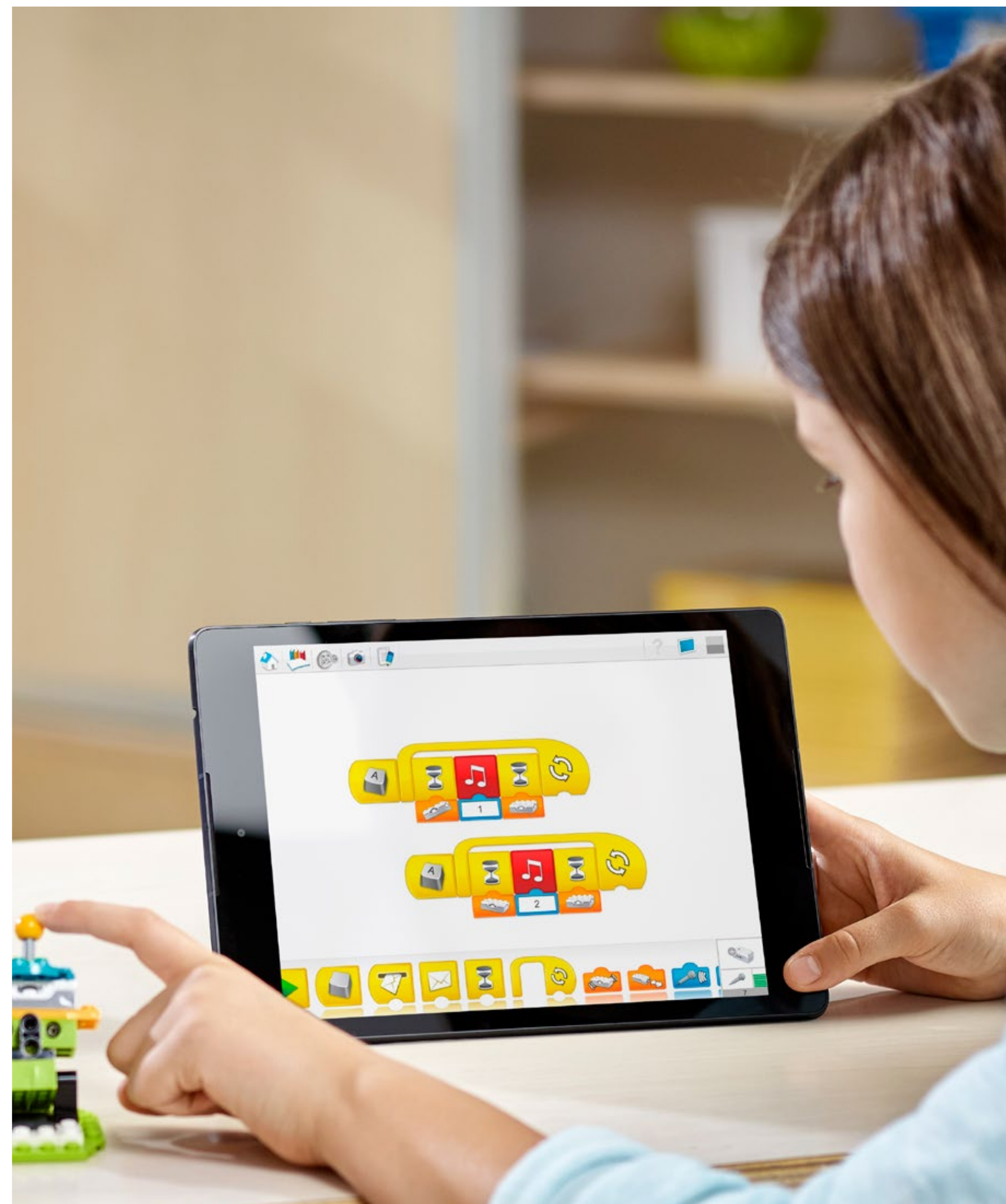
教育版レゴ® WeDo 2.0のプロジェクト学習でプログラミング的思考を発達させる

教育版レゴ® が自信を持って提供するこのプロジェクトは、小学校の授業で使用するために特別にデザインされており、子どもたちがプログラミング的思考力を身につけるのに役立ちます。

プログラミング的思考とは、日常的問題を解決するのに通常の学習で使える一連のスキルです。WeDo 2.0では、プロジェクトの各ステージを通してこうしたスキルが身につきます。各プロジェクトでは、教師のためにスキル向上ポイントを示していますが、自分や子どもたちにとって最も関連するプロジェクトに絞り込んでもよいでしょう。

WeDo 2.0のどのプロジェクトも、レゴのブロックを使ったアクティビティをアイコン式プログラミング言語と結びつけて、プログラミングの基本ルールを学びながら、問題の解決方法を見つけられるように工夫されています。

WeDo 2.0でコーディング・アクティビティを行いながらプログラミング的思考を発達させることで、子どもたちは自分の作品が動き出すのを見て笑顔になり、さらなる探求に意欲を燃やします。





コンピューターサイエンス、プログラミング的思考、コーディング

人類の創成期からある科学や技術という学習分野と比べると、コンピューターサイエンスは歴史こそはるかに浅いものの、科学や技術へのアプローチ方だけでなく、暮らし方にも影響を及ぼします。

コンピューターサイエンスは、科学 (Science)、技術 (Technology)、工学 (Engineering)、数学 (Mathematics) と特質を同じくする「STEM」学問分野です。

STEMのどの学問分野も物の見方と生涯役に立つ手法を発達させる機会となります。質問したり、解決法を練ったり、結果を伝えたりする能力はこうした手法の例です。

プログラミング的思考もこの手法に属し、問題解決に使える一般的な思考方法のひとつです。

プログラミング的思考はスキルの集まりとして説明することもでき、そのひとつが論理的思考です。論理的思考を考え出す作業を記述する際、「コード」や「コーディング」を使うことができます。

つまり、STEMの枠内でコーディングとは、プログラミング的思考を発達させるための一手段ととらえることができます。

STEM 学問分野

科学 (理科)、技術、工学、数学 (算数)、
コンピューターサイエンス

物の見方と生涯役に立つ手法を身につける

1. 質問し問題を解決する。
2. モデルを作る。
3. 試作品をデザインする。
4. 調べる。
5. データを分析し、理解することができる。
6. 計算論的思考を用いる。

- a.問題点を分ける
- b.発表・プレゼンテーション
- c.論理的思考
- d.試行錯誤
- e.規則性を見出す

7. 根拠を使って、議論を行うことができる。
8. 情報を取得・評価・伝達する。



プログラミング的思考とは

「プログラミング的思考」という表現を最初に提唱したのは数学者のシーモア・パパートですが、この概念を広めたジャネット・ウィング教授は、この概念を次のように定義しました。

「問題とその解決策を公式化して、解決方法を 何らかの形式で表せるようにする思考プロセス。情報処理ツールを使って効果的に実行可能である。」(ウィング教授、2011年)

プログラミング的思考は、様々な分野や状況で、日常的に利用されます。この思考に基づくスキルは、科学(理科)、工学、数学(算数)分野で発揮されます。こうしたスキルは以下のように定義できます。

問題点を分ける

分解とは、問題を細分して、解決策を見つけるプロセスを簡略化する能力です。分解することで、問題を他人に説明したり、タスクを分割するのが簡単になります。分解は一般化につながるのが通例です。

例：休暇にでかける場合、その準備(プロジェクト)は、航空チケットの予約、ホテルの予約、荷づくりなど、いくつかのサブタスクに分けることができます。

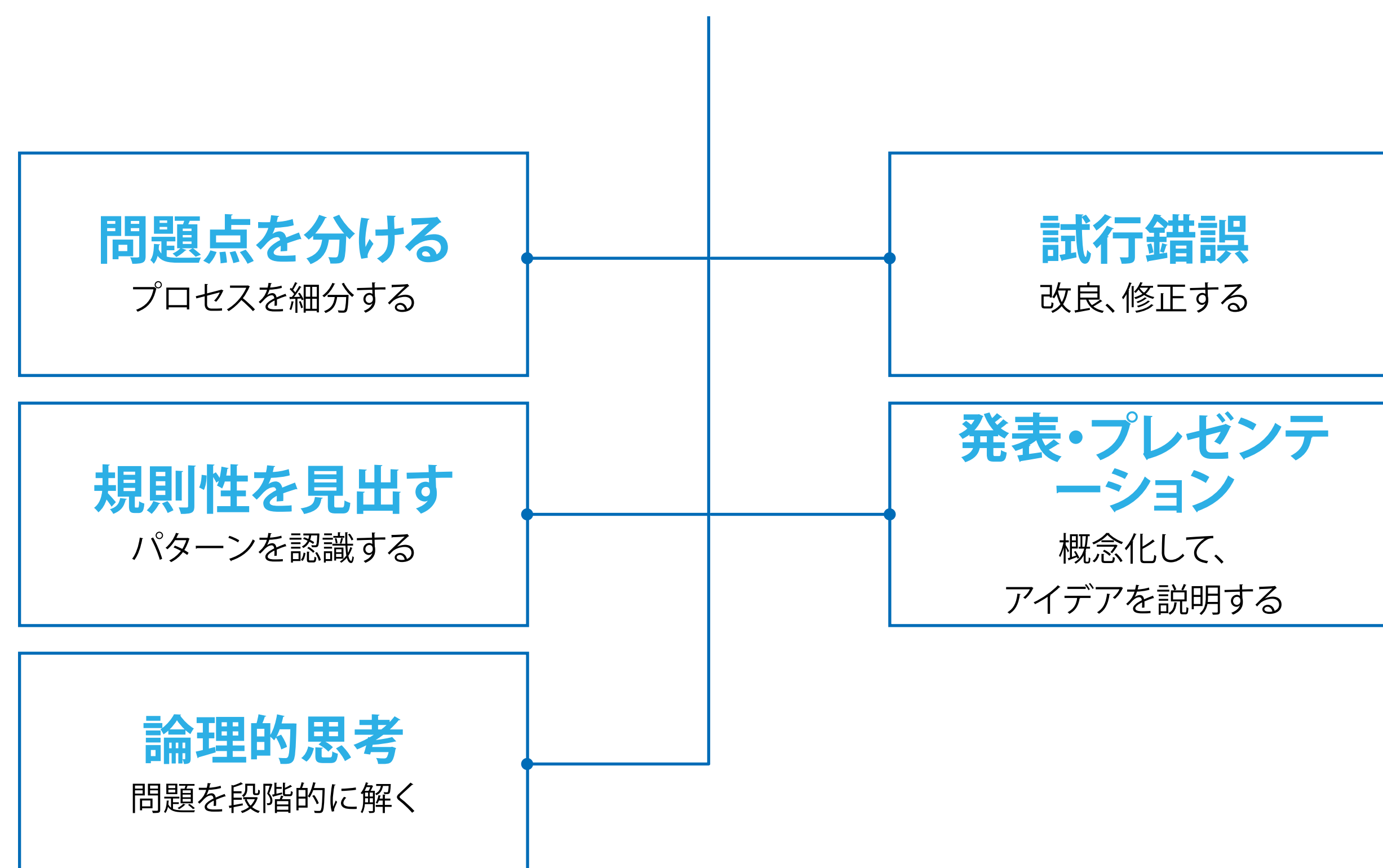
規則性を見出す(パターンの認識)

一般化とは既知または既視感のあるタスクの部分を認識する能力です。一般化の結果、アルゴリズムの考案が簡単になることがよくあります。

例：信号は同じ動作をずっと反復することで機能します。

プログラミング的思考

問題の解決策





プログラミング的思考とは

論理的思考

論理的思考とは、問題を解決するために、体系的な一連の手順を作り上げる能力です。

例1: レシピを参考にして料理する際は、食事を作るために、一連の手順に従います。

例2: コンピューターゲームをする際は、一連の操作をコード化して、コンピューターに動作を指示します。

試行錯誤または修正

これは、試作品がプランどおり動くかどうかを検証し、動かない場合、改良すべき点を特定する能力です。プログラム内の誤りを見つけて修正するためにコンピュータープログラマーが必ず行うプロセスでもあります。

例1: 料理する際は、時々味見をして調味料の量が適切かどうかを確認します。

例2: 文章のスペルミスや句読点の打ち忘れを探す際は、間違いを見つけそれを修正して、正しく読めるようにします。

発表・プレゼンテーション力

発表・プレゼンテーション力とは、重要でない部分を省略して問題や解決策を説明する能力です。アイデアを概念にする能力とも言えます。

例: 自転車を描写する際、一部の特徴に絞って説明します。タイプや色に触れるかもしれません。自転車に興味がある人にはもっと細かく説明するでしょう。



プログラミング的思考力を身につけるプロセス

工学的デザインプロセスの利用

問題の解決策を探す際、エンジニアが踏むのがデザインプロセスです。エンジニアは一連のステージを経て、解決策にたどり着きます。各ステージでは一部のスキルを用いるか習得します。こうしたスキルこそ我々が「プログラミング的思考スキル」と呼ぶものです。

WeDo 2.0で、子どもたちは類似のプロセスを踏みます。

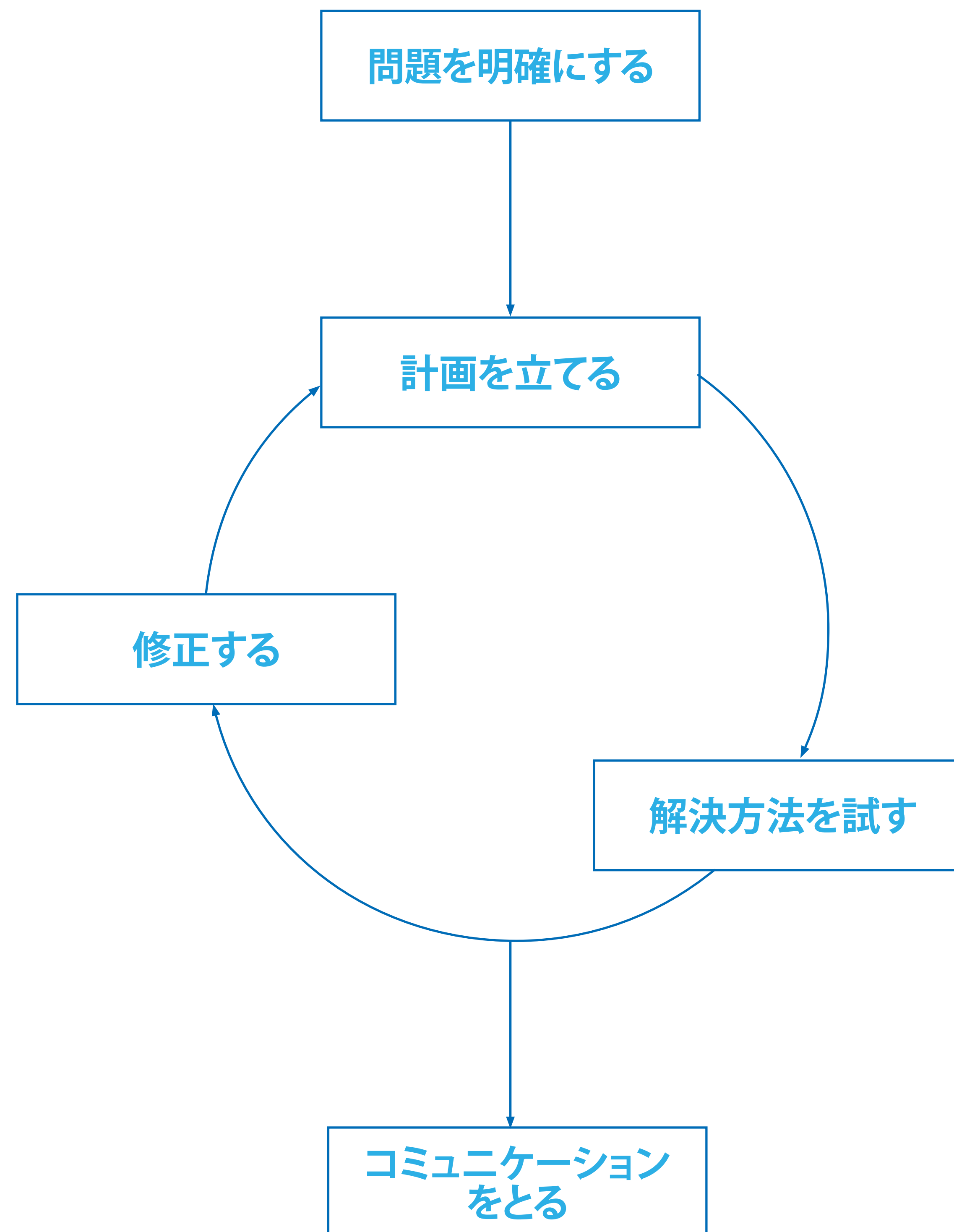
問題を明確にする

子どもたちにトピックが提示されます。このトピックは、改善したい問題や状況に発展していきます。細かい点がたくさんある問題もあります。そうした問題は細分すると解決しやすくなります。

問題を単純に定義し、何らかの達成基準を特定することで、子どもたちは「問題点を分ける」スキルを習得します。

質問形式にすると：

- 問題を自分で説明することができますか？
- 問題をうまく解決できたかどうかの判断方法を説明することができますか？
- 問題を細分して解決可能な単位に分解することができますか？





プログラミング的思考力を身につけるプロセス

計画を立てる

子どもたちは問題に対して色々な解決策をしばらく考えてから、そのうちの1つの詳しい実行プランを立てます。解決方法をつきとめるまでに通過する必要のある手順を明確にします。過去に見たことのあるようなタスクの部分を特定することで「規則性を見出す」スキルがみがかけられます。

質問形式にすると：

- プログラムの動作リストを作成することができますか？
- 使えそうなプログラムの一部を特定することができますか？
- プログラムの一部を再利用できますか？

試行する

次に子どもたちはそれぞれ、解決方法の仕上げに取り組みます。プロセスのこのステージではアイコン式プログラミング言語を使って、レゴ®のモデルを起動させます。アイデアをコード化する際、論理的思考力が身につきます。

質問形式にすると：

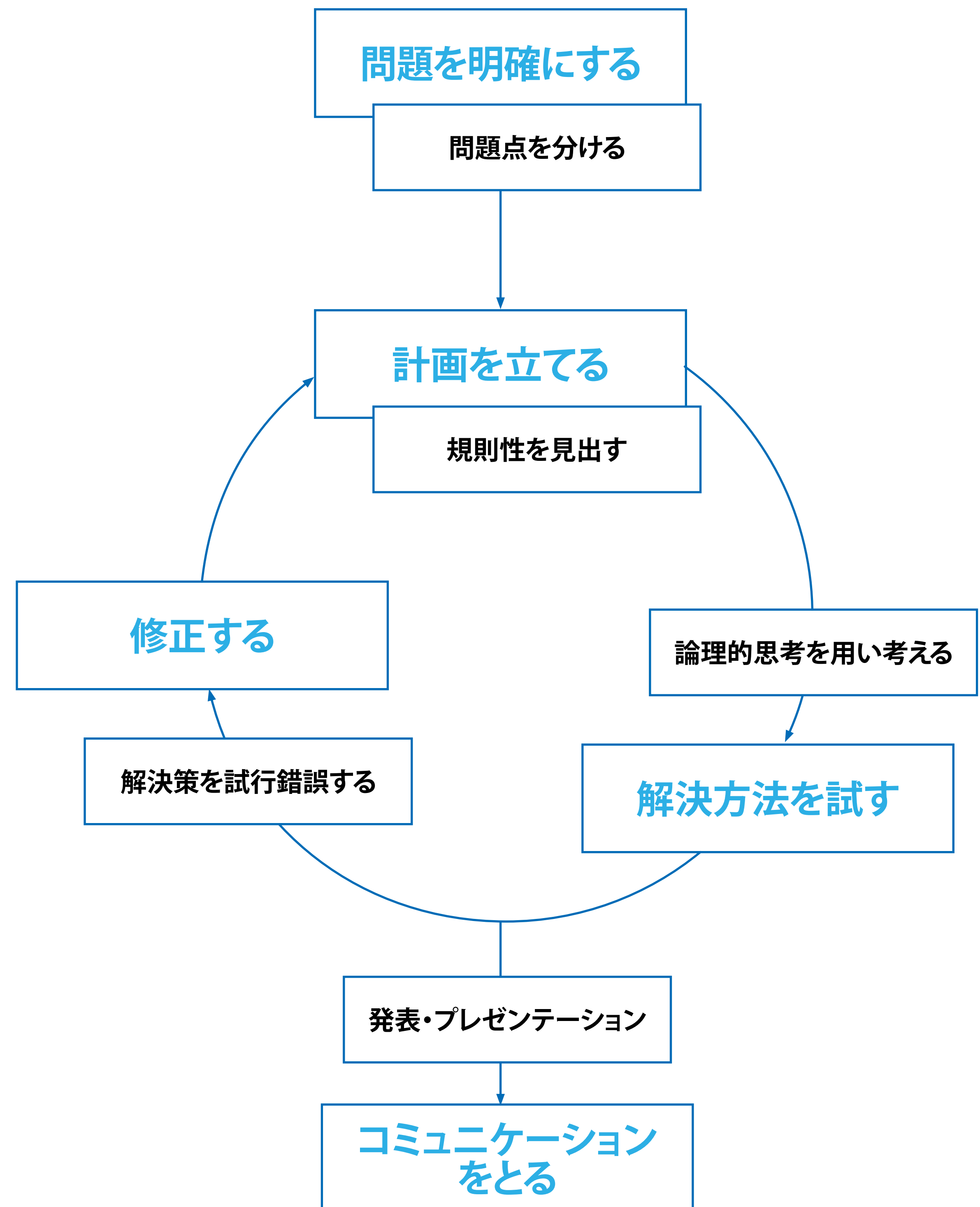
- プログラムの解決方法をプログラミングすることができますか？
- 連続したプログラム、ループ文、条件文を使うことができますか？

修正する

子どもたちは、プログラムとモデルが成功の判断基準を満たすかどうかに基づき自分たちの解決方法を評価します。評価スキルを使って、プログラムの一部を変更、調節、修正、改良する必要があるかどうかを判断します。

質問形式にすると：

- プログラムを反復(ループ)していますか？
- プログラムの問題を調節していますか？
- 解決方法が問題と関連付けられているかどうか判断することができますか？





プログラミング的思考力を身につけるプロセス

コミュニケーションをとる

子どもたちは、成功基準をどのように満たしているかを説明しながら、クラス全体に自分の解決方法の完成形を披露します。解決方法を適度に詳しく説明することで、抽象的概念とコミュニケーションスキルがみがかけられます。

質問形式にすると：

- 解決方法の最も重要な部分を説明していますか？
- 理解が深まるように十分な詳細を提示していますか？
- 成功基準に照らした評価をきちんと説明していますか？





コーディングを通じてプログラミング的思考を身につける

論理的思考を身につけるために、プログラミングの基本ルールを子どもたちに紹介します。解決方法を練る際、子どもたちは連続した動作と構造を体系化して、モデルを動かします。

以下は、WeDo 2.0のプログラミングの基本ルールのうち最もよく使われるものです。

1.出力

出力とは、作成されたプログラムによって制御されるもので、WeDo 2.0では例えば、サウンド、ライト、ディスプレイやモーターのオンとオフの切り替えが挙げられます。

2.入力

入力とは、コンピューターやデバイスが受信する情報のことで、数やテキストの値の形でセンサーを使用して入力可能です。例えば、距離などを検知または測定するセンサーは、こうした値をデジタル入力信号に変え、プログラムで使用できるようにします。

3.イベント(待機)

連続動作を実行する前に、何か起こるまで次の動作を待つようにプログラムに指示することができます。プログラムは一定時間、またはセンサーが何かを検知するのを待つことができます。

4.ループ

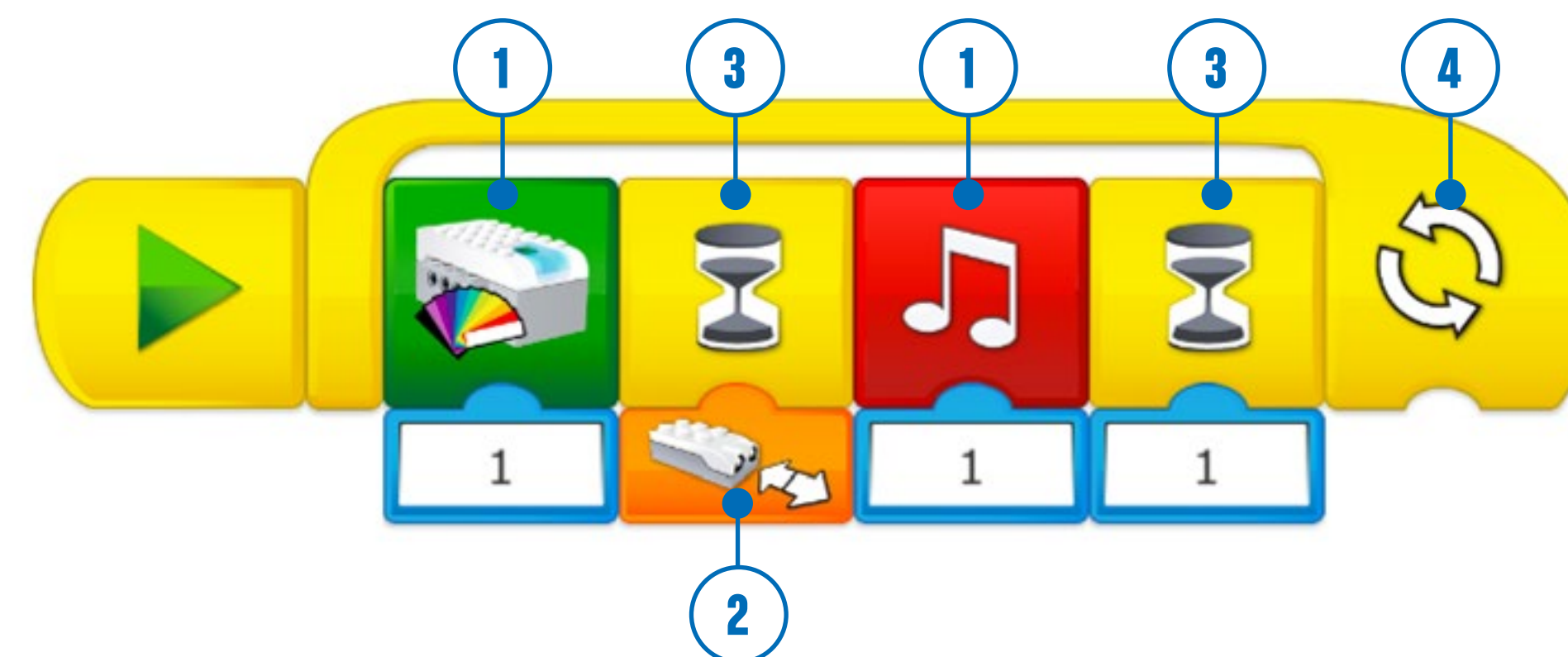
ずっとまたは一定時間反復されるように動作をプログラミングすることができます。

5.機能

機能とは動作の集まりであり、特定の状況でまとめて使用されます。例えば、ライトを点滅させるのに使用できるブロックの集まりは「点滅機能」と総称されます。

6.条件

条件とは、特定の状況でのみ実行される動作をプログラミングする場合に使用されます。プログラム内で条件を作成すると、その条件が満たされなければ、プログラムの一部は実行されなくなります。例えば、チルトセンサーが左方向にかたむくとモーターは始動し、右方向にかたむくとモーターは停止するという条件を付す場合、チルトセンサーが左方向にかたむかない限りモーターは始動せず、右方向にかたむかない限りモーターは停止しなくなります。



WeDo 2.0を使った 学習カリキュラム

レゴ エデュケーション® WeDo 2.0プロジェクトでは、レゴ®ブロックと次世代科学基準 (NGSS) とを融合させています。WeDo 2.0プロジェクトはすべて子どもたちがプログラミング的思考スキルを身につけられるようデザインされています。





プログラミング的思考のカリキュラムへの組み入れ

世界は変化しています。意識しようとしまいと、テクノロジーとコンピューターサイエンスは私たちの生活のほぼ全側面を方向づけています。子どもたちは急速にアクティブな市民となっており、適切なスキルを習得させることは国家の最優先課題のひとつにもなっています。

プログラミング的思考は世界的に普及しているスキルであり、テクノロジーと関連付けながら発達させる指導方法が主流となりつつあります。既に米国次世代科学スタンダード (NGSS) によって科学と工学分野に不可欠な実用知識として認識されているプログラミング的思考スキルは、世界各国のカリキュラムでも定着しています。

プログラミング的思考は、いくつかの団体の基準を根幹をなすようになってきました。こうした団体の代表的なものには、コンピューターサイエンス教員連盟 (CSTA)、テクノロジー教育国際協会 (ISTE)、Code.org、Computing at School (世界的に認められるコンピューター関連カリキュラムを統括する英国の連盟) があり、すべてプログラミング的思考スキルの発達に重点を置いたカリキュラムに沿った教育を推奨しています。

こうした重要スキルは、日常生活で出くわす状況に根差した、子どもたちの関心を引くアクティビティやプロジェクトを通じて習得できます。レゴ エデュケーション®ではこうした発達をサポートする目的で、プログラミング的思考プロジェクト専用のシリーズを科学プロジェクトに加え、既に WeDo 2.0 で発売しています。



基礎プロジェクトの概要

1. 月面基地

このプロジェクトでは、ロボットで月面基地を組み立てる方法を考案します。

2. 物をつかむ

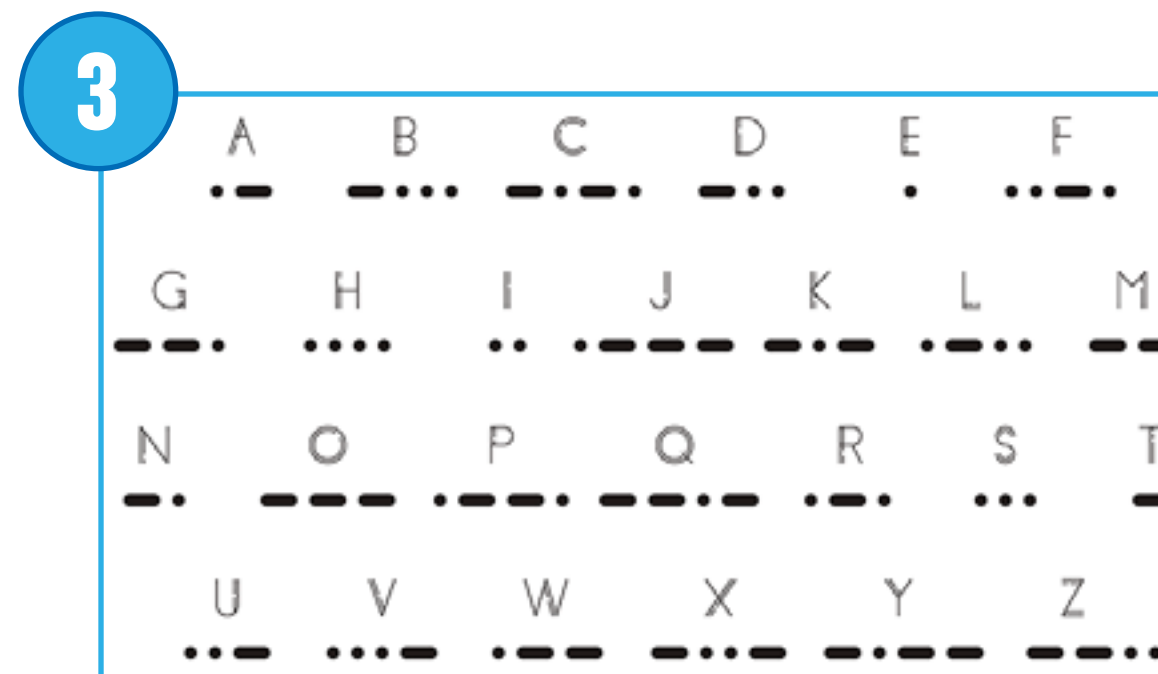
このプロジェクトでは、周囲にある小さな物を義肢で動かす方法を考案します。

3. メッセージの送信

このプロジェクトでは、パターンにまとめられた信号システムを使って情報をやり取りする方法を考案します。

4. 火山警告

このプロジェクトでは、科学探検を導くために、火山活動の監視方法を改善するためのデバイスを考案します。





発展プロジェクトの概要

5. 検査

このプロジェクトでは、センサーで動作を操作しながら、ロボットが狭い空間を検査できるようにする方法を考案します。

6. 感情をデザイン

このプロジェクトでは、人間と接触する際、ロボットが好意的な感情を示す方法を考案します。

7. 都市の安全

このプロジェクトでは、都市の安全を強化する方法を考案します。

8. 動物の五感

このプロジェクトでは、環境と共存するために動物が五感を用いる方法のモデルを組み立てます。

5



6



7



8





プログラミング的思考力を身につけるプロセスフロー (案)

各プロジェクトの実施順序に決まりはありません。どのプロジェクトもプログラミング的思考スキルの発達機会となりますが、自分と子どもたちに最も関連するプロジェクトに集中するよう、独自のプランを立てることもできます。以下に挙げる一例では、取り上げるプログラミング概念の複雑度を基にプランが組まれています。

プロジェクトの入門

45分のレッスン2コマを使って、WeDo 2.0を子どもたちに紹介します。

レッスン1、科学探査機:マイロ

レッスン2、マイロのモーションセンサー、チルトセンサー、共同作業の組み合わせ

基礎プロジェクト

45分のレッスン2コマを使って、子どもたちに一連の動作をプログラミングしてもらいます。

レッスン3、月面基地(「調べる」と「組み立てる」ステージ)

レッスン4、月面基地(「テストする」と「発表する」ステージ)

45分のレッスン2コマを使って、子どもたちにセンサー(入力)を使ってもらいます。

レッスン5、物をつかむ(「調べる」と「組み立てる」ステージ)

レッスン6、物をつかむ(「テストする」と「発表する」ステージ)

45分のレッスン2コマを使って、子どもたちにセンサー(入力)、ループ、並列プログラミングを扱ってもらいます。

レッスン7、メッセージの送信(「調べる」と「組み立てる」ステージ)

レッスン8、メッセージの送信(「テストする」と「発表する」ステージ)

45分のレッスン2コマを使って、子どもたちに条件の概念を紹介し、ほかのプログラミングの基本ルールとの組み合わせ方を説明します。

レッスン9、火山警告(「調べる」と「組み立てる」ステージ)

レッスン10、火山警告(「テストする」と「発表する」ステージ)

発展プロジェクト

45分のレッスン2~3コマを使って、発展プロジェクトのいずれかの例を参考にして、独自のプロジェクトを作ります。このプロジェクトには、プログラミングの基本ルールと、基礎プロジェクトで習得したプログラミング的思考スキルをすべて織り込みます。



プログラミング的思考力を身につけるプロセスフロー (案)

プロジェクトの入門

子どもたちに WeDo 2.0 を紹介する



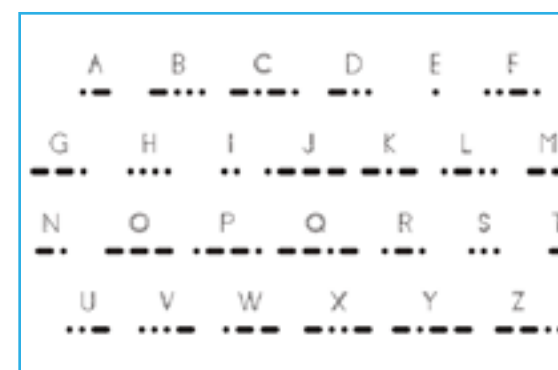
45分



45分

基礎プロジェクト - メッセージの送信

子どもたちはセンサー (入力)、ループ、並列プログラミングを使います。



集中レッスンプローを使用
2 x 45 分

基礎プロジェクト - 月面基地

子どもたちは一連の動作をプログラミングします



集中レッスンプローを使用
2 x 45 分

基礎プロジェクト - 火山警告

条件などのプログラミングの基本ルールを紹介します。



集中レッスンプローを使用
2 x 45 分

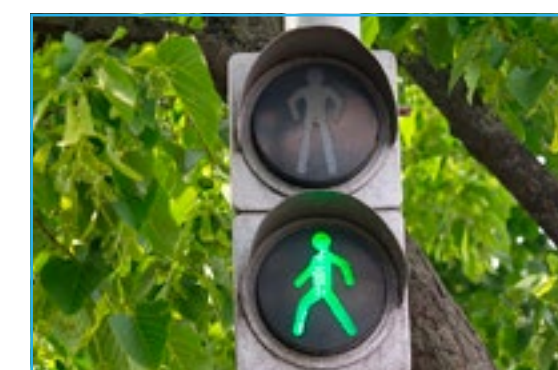
基礎プロジェクト - 物をつかむ

子どもたちはセンサー (入力) を使います。



集中レッスンプローを使用
2 x 45 分

発展プロジェクト





基礎プロジェクトのカリキュラムの概要

| | ① 月面基地 | ② 物をつかむ | ③ メッセージの送信 | ④ 火山警告 |
|-------|--|---|---|---|
| 生物分野 | | | | |
| 地学分野 | | | | 4-ESS3-2. |
| 物理分野 | | | 4-PS4-3. | |
| 工学、技術 | K-2-ETS1-3. 3-5-ETS1-1. 3-5-ETS1-2. 3-5-ETS1-3. | 3-5-ETS1-1. 3-5-ETS1-2. 3-5-ETS1-3. | 3-5-ETS1-1. 3-5-ETS1-2. 3-5-ETS1-3. | 3-5-ETS1-1. 3-5-ETS1-2. 3-5-ETS1-3. |



発展プロジェクトのカリキュラムの概要

| | 5 検査 | 6 感情をデザイン | 7 都市の安全 | 8 動物の五感 |
|-------|--|--|--|--|
| 生物分野 | | | | 4-LS1-2. |
| 地学分野 | | | | |
| 物理分野 | | | | |
| 工学、技術 | K-2-ETS1-3. 3-5-ETS1-1. 3-5-ETS1-2. 3-5-ETS1-3. | K-2-ETS1-3. 3-5-ETS1-1. 3-5-ETS1-2. 3-5-ETS1-3. | K-2-ETS1-3. 3-5-ETS1-1. 3-5-ETS1-2. 3-5-ETS1-3. | K-2-ETS1-3. 3-5-ETS1-1. 3-5-ETS1-2. 3-5-ETS1-3. |



WeDo 2.0 を使った学習目標:その1

生物分野

- 2-LS2-1.** 植物の成長に、太陽の光と水が必要であることを証明する調査を計画し、実施することができる。
- 2-LS2-2.** 植物の種子の飛散や、受粉を助ける生き物の役割を再現する、簡単なモデルを制作することができる。
- 2-LS4-1.** 異なる場所で、植物と動物を観察し、種の多様性を比較することができる。

物理分野

- 2-PS1-1.** 異なる物質について、観察できる性質をまとめ、それを元に、物質を分類するための調査を計画し、実施することができる。
- 2-PS1-2.** 異なる物質を試験して得られたデータを分析し、使用目的に最も適した性質の物質を特定することができる。
- 2-PS1-3.** 観察に基づき、少数の要素で構成された物質を分解して、完全に違う物体に組み替える方法を、証拠を示しながら説明することができる。
- 2-PS1-4.** 加熱または冷却によって生じる変化には、元に戻せるものと、戻せないものがあることを、証拠を示しながら説明することができる。

地学分野

- 2-ESS1-1.** 複数のソースから収集した情報を用い、地球上で発生する事象には短期的なものや長期的なものがあることを証拠を示しながら説明することができる。
- 2-ESS2-1.** 水や風による地形の変化を、小さくしたり、無くしたりする複数の解決策を比較することができる。
- 2-ESS2-2.** 1つの地域の地形や地質、水域を再現するモデルを制作することができる。
- 2-ESS2-3.** 地球上の水がどこにあるかを調べ、水が固体でも液体でも存在していることを理解することができる。

工学

- K-2-ETS1-1.** 人々を変えたいと思っている状況について調べ、観察し、情報を収集して、何らかの新しい物体やツールの開発、または改善によって解決できる簡単な問題を特定することができる。
- K-2-ETS1-2.** 問題の解決をするために、その形がどのように働きを助けるかについて、説明するために、簡単なスケッチ、図面、または実際のモデルを制作することができる。
- K-2-ETS1-3.** 1つの問題を解決するために制作した2種類のツールを試験し、データを分析してそれぞれの長所と短所を比較することができる。



WeDo 2.0を使った学習目標:その2

物理分野

3-PS2-1. 物の運動において、力のバランスがとれた時と、力のバランスがとれていない時の現象について、根拠を得るために、実験を計画し実施することができる。

3-PS2-2. 物体の運動に対する観察や計測を行い、規則性を用いて、未来の運動を予測できる理論を提示することができる。

3-PS2-3. 互いに接していない、2つの物体の電氣的または、磁氣的相互作用について調べ、原因と効果の関係性を見つけることができる。

3-PS2-4. 磁石の科学的概念を応用することによって解決できる、簡単なデザイン上の問題を定義することができる。

地学分野

3-ESS2-1. 特定の季節に、一般的に見られる気象について、データを表や図で表すことができる。

3-ESS2-2. 世界の異なる地域の気象に関するデータを収集し、説明することができる。

3-ESS3-1. 気象災害の影響を抑えることができる解決策の利点について、自分の意見をまとめることができる。

工学

3-5-ETS1-1. 素材や時間、費用の制限など、具体的な成功の条件があり、特定のニーズが反映された、デザイン上の簡単な問題を定義することができる。

3-5-ETS1-2. 1つの問題に対する考え得る解決策を製作し、成功の条件や制限をどれほど満たせているか比較することができる。

3-5-ETS1-3. 変数が制御され、不具合点を把握することができる公平な試験を行い、モデルまたは試作品の改善点を見つけすることができる。

生物分野

3-LS2-1. 一部の動物は、生存のために群れを形成するという考えを支持する理論をまとめることができる。

3-LS4-1. 化石から集められたデータを分析・解釈し、過去に存在した生命体と当時の環境を示す証拠を提示することができる。

3-LS4-3. 特定の生息地で、種によって生存率の高低に差があり、生存できない種もあるとする議論を、証拠を使ってまとめることができる。

3-LS4-4. 環境の変化と、そこに住む動植物の性質の変化によって起こる問題に対する解決策の利点に関する主張を書くことができる。

3-LS1-1. 種の寿命はそれぞれに異なるが、誕生、成長、繁殖、死という変化は共通することを説明するモデルを制作することができる。

3-LS3-1. データを分析・判断し、植物や動物には親から受け継いだ特質があり、同一の種においても、これらの特徴には個体間でばらつきがあることを根拠を示しながら説明することができる。

3-LS3-2. 生物の特質は、環境に影響されるという理論を支持する考えを、根拠を示しながら説明することができる。

3-LS4-2. 同一の種における特質のばらつきが、種の存続、パートナー探し、繁殖において有利であるとする考えを、根拠を示しながら説明することができる。



WeDo 2.0を使った学習目標:その3

物理分野(エネルギー)

- 4-PS3-1.** 物体の速度が、その物体の持つエネルギーに比例するとする理論を、例をあげて説明することができる。
- 4-PS3-2.** 観察を通して、エネルギーを音、光、熱、電流に変換させることができることを例をあげて説明することができる。
- 4-PS3-3.** 物体同士が衝突する際に生じるエネルギーの変化について考え、結果を予測することができる。
- 4-PS3-4.** 科学的概念を応用して、エネルギーを1つの形から別の形へ変化させる装置のデザイン、試験、改善を行うことができる。
- 4-ESS3-1.** エネルギーと燃料が、天然資源から得られており、その使用が、まわりの環境に影響を与えているという事実を調べ、自分の考えを発表することができる。

技術(構造、機能、情報処理)

- 4-PS4-2.** 物体によって反射された光が、眼に入ることで、その物体を見ることができるしくみを、説明するモデルを制作することができる。
- 4-LS1-1.** 植物と動物には、生存、成長、行動、繁殖を可能にする内部構造と外部構造があることを調べ、考えをまとめることができる。
- 4-LS1-2.** 動物が、五感によって様々な情報を得て、頭脳で処理し、それに対する様々な行動をとることをモデルを用いて説明することができる。

物理分野(波:波と情報)

- 4-PS4-1.** 振幅と波長における規則性と、波が物体を動かすことができることを示す波モデルを制作することができる。
- 4-PS4-3.** 波の規則性を用いて、情報を伝達する解決策を複数制作し、比較することができる。

地学 分野:

- 4-ESS1-1.** 岩石の形成における規則性や岩の層に存在する化石などから、時間の経過に伴う地形の変化に関する理論の根拠を見つけることができる。
- 4-ESS2-1.** 観察や測定を行い、風化作用または水、氷、風、植物による浸食速度を示す証拠を示すことができる。
- 4-ESS2-2.** 地図から得られた情報を分析・判断して、土地の性質における規則性を説明することができる。
- 4-ESS3-2.** 地球上の自然現象による人間への影響を抑えるための解決策を、複数制作し、比較することができる。

工学

- 3-5-ETS1-1.** 素材や時間、費用の制限など、具体的な成功の条件があり、特定のニーズや要望が反映された、デザイン上の簡単な問題を定義することができる。
- 3-5-ETS1-2.** 1つの問題に対する考え得る解決策を制作し、成功の条件や制限をどれほど満たせているか比較することができる。
- 3-5-ETS1-3.** 変数が制御され、不具合点を把握することができる公平な試験を行い、モデルまたは試作品の改善点を、見つけることができる。



科学的・工学的学習方法による基礎プロジェクトのカリキュラム概要

| | ① 月面基地 | ② 物をつかむ | ③ メッセージの送信 | ④ 火山警告 |
|--|-----------|------------|---------------|-----------|
| 手法 1: 質問をし問題を見出せる | ● | ● | ● | ● |
| 手法 2: モデルを作り、使用することができる | | | | |
| 手法 3: 調査を計画し、実行することができる | | | | |
| 手法 4: データを分析し、理解することができる | | | | |
| 手法 5: 数学と計算論的思考を使用することができる | ● | ● | ● | ● |
| 手法 6: 説明を考え、解決策をデザインすることができる | ● | ● | ● | ● |
| 手法 7: 根拠を使って、議論を行うことができる | ● | ● | ● | ● |
| 手法 8: 情報の取得・評価・伝達を行うことができる | ● | ● | ● | ● |



科学的・工学的手法による発展プロジェクトのカリキュラム概要

| | 5 検査 | 6 感情をデザイン | 7 都市の安全 | 8 動物の五感 |
|--|---------|--------------|------------|------------|
| 手法 1: 質問をし問題を見出せる | ● | ● | ● | ● |
| 手法 2: モデルを作り、使用することができる | | | | ● |
| 手法 3: 調査を計画し、実行することができる | | | | |
| 手法 4: データを分析し、理解することができる | | | | |
| 手法 5: 数学と計算論的思考を使用することができる | ● | ● | ● | ● |
| 手法 6: 説明を考え、解決策をデザインすることができる | ● | ● | ● | |
| 手法 7: 根拠を使って、議論を行うことができる | ● | ● | ● | ● |
| 手法 8: 情報の取得・評価・伝達を行うことができる | ● | ● | ● | ● |



CSTA 基準に基づく基礎プロジェクトおよび発展プロジェクトのカリキュラム概要

| 識別記号 | | CSTA初期中等教育向けコンピューターサイエンス教育基準(暫定) | 1 月面基地 | 2 物をつかむ | 3 メッセージの送信 | 4 火山警告 | 5 検査 | 6 感情をデザイン | 7 都市の安全 | 8 動物の五感 |
|------|-----------|---|-----------|------------|---------------|-----------|---------|--------------|------------|------------|
| K-2 | 1A-A-3-7 | シーケンス、単純なループを含むアルゴリズム(段階別の指示一式)を個人単位とチーム単位の両方で構築、実行してタスクを完成させる(コンピューターデバイスの使用の有無を問わない)。 | ● | ● | ● | ● | ● | ● | ● | ● |
| K-2 | 1A-A-6-8 | シーケンスや単純なループを含むアルゴリズムをコンピューターデバイスを使ってまたは使わずに解析、デバッグ(修正)する。 | ● | ● | ● | ● | ● | ● | ● | ● |
| K-2 | 1A-C-7-9 | コンピューター計算デバイスの操作ソフトウェアを特定し、使用する(アプリを使って画面に絵を描く、ソフトウェアを使って物語を書いたりロボットを操作したりするなど)。 | ● | ● | ● | ● | ● | ● | ● | ● |
| K-2 | 1A-C-7-10 | 一般的なコンピューターデバイスとコンポーネントの機能を適切な用語で呼び、説明する(デスクトップコンピューター、ノートブックコンピューター、タブレット、モニター、キーボード、マウス、プリンターなど)。 | | | | | | | | |
| K-2 | 1A-C-6-11 | ハードウェアやソフトウェアの使用中に起こり得る単純な問題を正確な用語で特定する(アプリやプログラムが予想どおり機能しない、サウンドが出ない、デバイスの電源がつかないなど)。 | ● | ● | ● | ● | ● | ● | ● | ● |
| K-2 | 1A-D-7-12 | 時間をかけてデータを収集したら、表やグラフにまとめて予想を立てる。 | | | | | | | | |
| K-2 | 1A-D-4-13 | コンピューターデバイスを使って、情報を保存、検索、呼び出し、変更、削除して、保存した情報をデータとして定義する。 | | | | | | | | |
| K-2 | 1A-D-4-14 | パターンや必須要素を特定するため、オブジェクトやプロセスのモデルを作成する(水のサイクル、蝶のライフサイクル、季節での天気の移り変わりなど)。 | ● | ● | ● | ● | ● | ● | ● | ● |



CSTA 基準に基づく基礎プロジェクトおよび発展プロジェクトのカリキュラム概要

| 識別記号 | | CSTA初期中等教育向けコンピューターサイエンス教育基準(暫定) | 1 月面基地 | 2 物をつかむ | 3 メッセージの送信 | 4 火山警告 | 5 検査 | 6 感情をデザイン | 7 都市の安全 | 8 動物の五感 |
|------|----------|--|-----------|------------|---------------|-----------|---------|--------------|------------|------------|
| 3-5 | 1B-A-2-1 | 問題を解決する際、プログラムのデザインサイクルの枠内で、共同作業的手段を適用する。 | ● | ● | ● | ● | ● | ● | ● | ● |
| 3-5 | 1B-A-7-2 | 借用したアイデアを応用する場合(他人が制作した写真や音楽を使用する場合やプログラミングプロジェクトを編集する場合など)、適切な引用や記録を用いる。 | ● | ● | ● | ● | ● | ● | ● | ● |
| 3-5 | 1B-A-5-3 | 個人単位と多彩な共同チーム単位の両方で、反復的デザインプロセスの構成要素となるプランを制作する(ストーリーボード、フローチャート、擬似コード、ストーリーマップなど)。 | ● | ● | ● | ● | ● | ● | ● | ● |
| 3-5 | 1B-A-5-4 | 問題を解決するため、または創造的な表現として、プログラムを個人単位とチーム単位(ペアプログラミング)の両方で構築する。プログラムには、シーケンス、イベント、ループ、条件、並列、変数を含み、ブロック型の視覚的プログラミング言語かテキスト型言語を使用する。 | ● | ● | ● | ● | ● | ● | ● | ● |
| 3-5 | 1B-A-5-5 | 数学的計算を用いて、変数として保存されている値を変更する。 | | | | ● | | | | |
| 3-5 | 1B-A-3-6 | 個人単位またはチーム単位で、長文の問題を短い問題に分解する。 | ● | ● | ● | ● | ● | ● | ● | ● |



CSTA 基準に基づく基礎プロジェクトおよび発展プロジェクトのカリキュラム概要

| 識別記号 | | | CSTA初期中等教育向けコンピューターサイエンス教育基準(暫定) | 1 月面基地 | 2 物をつかむ | 3 メッセージの送信 | 4 火山警告 | 5 検査 | 6 感情をデザイン | 7 都市の安全 | 8 動物の五感 |
|------|-----------|--|----------------------------------|-----------|------------|---------------|-----------|---------|--------------|------------|------------|
| 3-5 | 1B-A-3-7 | シーケンス、ループ、条件を含むアルゴリズム(段階別の指示一式)を個人単位とチーム単位の両方で構築、実行してタスクを完成させる(コンピューターデバイスの使用の有無を問わない)。 | | ● | ● | ● | ● | ● | ● | ● | ● |
| 3-5 | 1B-A-6-8 | シーケンス、イベント、ループ、条件、並列、変数を含むアルゴリズムを解析、デバッグ(修正)する。 | | ● | ● | ● | ● | ● | ● | ● | ● |
| 3-5 | 1B-C-7-9 | コンピューターシステムが動作する仕組みをモデル化する(説明:入力、出力、プロセッサ、センサー、ストレージといったコンピューターシステムの基本的要素のみを含むこと)。 | | | | | | | | | |
| 3-5 | 1B-C-7-10 | コンピューターデバイスの内部・外部コンポーネントに適切な用語で呼び名を付け、それぞれの関係、機能、制約を説明する。 | | | | | | | | | |
| 3-5 | 1B-C-6-11 | ハードウェアやソフトウェアの使用中に起こり得る単純な問題を正確な用語を使って特定し、そうした問題の解決策を実践する(デバイスの再起動、電源の点検、ネットワーク接続の点検、アプリの再起動など)。 | | | | | | | | | |
| 3-5 | 1B-D-5-12 | 特定コンセプトに関連する特質と行動をモデル化した人工物をコンピューターで制作する。例えば、太陽系、植物のライフサイクルなど。 | | ● | ● | ● | ● | ● | ● | ● | ● |
| 3-5 | 1B-D-5-13 | コンピューター操作(並び替え、合計、平均、表、グラフなど)を使ってしながら質問に回答し、クラス全体か個人が収集したデータを分析する。 | | | | | | | | | |



CSTA 基準に基づく基礎プロジェクトおよび発展プロジェクトのカリキュラム概要

| 識別記号 | | CSTA基準(暫定) | 1 月面基地 | 2 物をつかむ | 3 メッセージの送信 | 4 火山警告 | 5 検査 | 6 感情をデザイン | 7 都市の安全 | 8 動物の五感 |
|------|-----------|---|-----------|------------|---------------|-----------|---------|--------------|------------|------------|
| 3-5 | 1B-D-4-14 | 数値を使って、コンピューターの非数的概念を表す。例えば、バイナリ、ASCII、RGB などのピクセル特質。 | | | ● | | | | | |
| 3-5 | 1B-I-7-15 | コンピューターやコンピューターの利用が日常生活に浸透することによる利点と弊害を見極め、説明する。例えば、ビデオや音楽ファイルのダウンロード、電子機器、無線インターネット、モバイルコンピューターデバイス、GPS システム、装着式コンピューターデバイスなど。 | | | | | | | | |
| 3-5 | 1B-I-7-16 | コンピューターの利用が社会にどのような影響を与えることができるかを例示する。また、社会的価値によってコンピューターの利用方法が決まる例も挙げる。 | | | | | | | | |
| 3-5 | 1B-I-1-17 | 多様な角度、同期、非同期的にプロジェクトを追求、比較しながら改良を図る。 | | | | | | | | |
| 3-5 | 1B-I-1-18 | コンピューターデバイスの利便性を全ユーザーにとって高めることができる方法について様々な意見を出し合う。 | | | | | | | | |
| 3-5 | 1B-I-1-19 | コンピューターデバイスやネットワークの利用に関する問題を説明する。例えば、アカウントの不正利用を防ぐためのログアウト、ネットでのいじめ、個人情報保護、オーナーシップなどの問題。 | | | | | | | | |
| 3-5 | 1B-N-7-20 | 安全度の高いパスワードを例示し、そうしたパスワードを使用すべき理由を説明し、個人パスワードの適切な使用と保護を実践する。 | | | | | | | | |
| 3-5 | 1B-N-4-21 | ネットワーク内デバイス間で、特定のルールに従った(送信者から受信者への)メッセージのやり取り方法をモデル化する。 | | | ● | | | | | |

プログラミング 的思考をする

WeDo 2.0プロジェクトを用いて子どもたちの成長を監督・評価する方法は数多くあります。

このセクションでは次のツールを評価に役立てることができます。

- 記録ページ
- 自己評価文
- 実例記録シート
- 観察基準シート





子どもたち自身が行う評価

記録ページ

各プロジェクトの最後に、子どもたち自身が、自分の学習の成果をまとめる記録を作成するように、メッセージが書かれています。完成度の高い科学レポートを作成するには、重要なポイントがあります。

- 様々な種類のメディアを使って記録する。
- プロセスの、全ステップを記録する。
- 記録に記載する内容をまとめ、入力する時間をとる。

最初の記録は、完成度はあまり高くないかもしれませんが、学習を重ねるごとに、上達していくでしょう。以下の方法で子どもたちをサポートします。

- フィードバックを提供し、時間を十分に与え、記録のどこを変えればさらに良くなるかをつきとめる。
- 子どもたち同士で、完成した記録を共有させる。自分たちの発見した科学的事実について、情報を交換することで、科学者の実際の仕事を体感する。

自己評価文

プロジェクトの最後に、自分の作品を振り返る時間を設けましょう。次のページを使って、プロジェクトの振り返りと、次のプロジェクトの目標について考えさせてください。





学習者用自己評価シート

名前:

クラス:

プロジェクト:

指示: 自分の出来栄を示すブロックを丸で囲みましょう。ブロックが大きいほど、よくできたことになります。

| | | | | |
|------------------------------|--|--|--|--|
| 質問や問題を明確にした。 | | | | |
| レゴ®モデルを組み立てて、解決方法をプログラミングした。 | | | | |
| 解決方法をテストして、改良を加えた。 | | | | |
| やったことを記録して、自分のアイデアを共有した。 | | | | |

プロジェクトの反省

よくできた点:

次回は改善したい点:



教師による学習評価

子どもたちの科学的・工学的・プログラミング的思考力を育むには、時間とフィードバックが必要です。学習過程において失敗がプロセスの一部として考えられるように、評価は、何がよくできたか、何を改善しなければいけないかを子どもたちが理解できるようにするものでなければなりません。問題解決を中心とした学習では、成功か失敗かは重要ではありません。大切なのは、積極的に学び、継続的に学習に対するイメージを膨らませて実際に試してみることにあります。

スキルを習得させるために子どもたちにフィードバックを伝えるには、色々な方法を用いることができます。WeDo 2.0プロジェクトの各ステージでは、付属する評価時の指示の例を参考にしながら次のことができます。

- 子どもたちの行動、反応、方策を観察する
- 思考プロセスについて質問する

グループ活動が多いため、チームと個人の両方に対してフィードバックを提供できます。

実例記録シート

実例記録シートでは、それぞれの子どもの学習状況を記録します。必要に応じて、次のページにあるテンプレートを使って、子どもたちに学習進度についてのフィードバックをしましょう。




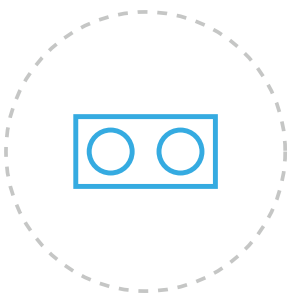
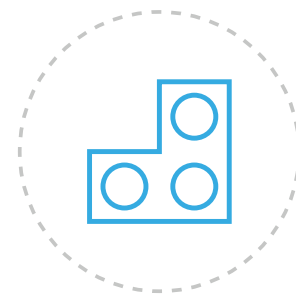
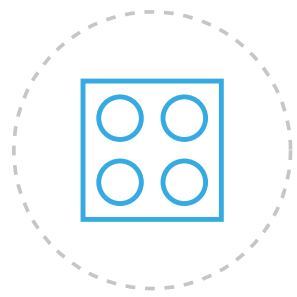


実例記録シート

名前:

クラス:

プロジェクト:

| 1.学習のはじめ | 2.学習の途中 | 3.学習のおわり | 4.学習終了後 |
|---|---|---|---|
|  |  |  |  |

注意事項:



教師による学習評価

観察基準シート

基礎プロジェクトに、ルーブリックのシートの使用例が載っています。それぞれの子どもまたは、チームに対して、次のように、ルーブリックのシートを利用することができます。

- 各ステップにおける、子どもたちの学習成果を評価する。
- 子どもたちの成長を助ける建設的なフィードバックをする。

基礎プロジェクト内にあるルーブリックのシートは、必要に応じて使い分けることができます。ルーブリックのシートは、下のような学習経過に基づいています。

1.学習のはじめ

学習内容に対する知識を持ち、内容を理解し、応用しようとする意識があり、常に学習に対して考えようとしている。

2.学習の途中

基本的な知識(例えばボキャブラリーなど)があるが、内容に対する知識を応用したり、新しい概念に対して理解することができない。

3.学習のおわり

提示される内容とその概念に対する確固とした理解があり、学習する項目、内容、または概念を適切に説明することができる。議論に参加したり、学習した内容以外に応用させることはできない。

4.学習終了後

学習した概念や、考えを次の段階に進め、概念を他の状況に応用したり、情報を組み合わせて、応用し、発展させて議論に用いることができる。これには自分の考えの発展も含める。

▶ おすすめ

次のページにあるルーブリックのシートを使って、子どもたちの成長の記録をつけることもできます。





観察基準シート

| | | | | | |
|------|--|------------|-------|-------|------|
| クラス: | | プロジェクト: | | | |
| 名前 | | 科学的な見方・考え方 | | | |
| | | 調べる | 組み立てる | テストする | 発表する |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

次のページで説明するルーブリックのシートの説明に従って、使用してください。(1) 学習のはじめ、(2) 学習の途中、(3) 学習のおわり、(4) 学習終了後



プロジェクトの各ステージの評価 - 一般的評価基準

この評価基準シートには一般的なフィードバックを1～4段階で記載します。プロジェクトの各ステージの最後に使用します。

調べる

このステージでは、子どもたちが質問や返答をしながら話し合いに積極的に参加しているかどうか、および問題の理解度に関連するフィードバックを記載します。

1. 質問に返答できない、または話し合いに十分参加できない。
2. 助けを借りれば、質問に回答できる。または話し合いに十分参加できる。
3. 質問にきちんと返答し、クラスでの話し合いに参加できる。
4. クラス全体での話し合いに、説明も交えて参加できる。

テストする

「テストする」ステージでは、すべての子どもたちが、チームの一員として作業し、自分の解決方法の正当性を説明し、「調べる」ステージで得た情報を活用できているか確認してください。

1. チームの一員として作業を行い、解決方法の正当性を説明し、取得した情報を活用することができない。
2. チームの一員として作業し、助言または助けを借りて情報を収集・活用し、解決策の正当性を説明できる。
3. チームの一員として作業し、チームの話し合いに貢献し、解決方法の正当性を説明し、学習内容についての情報を収集・活用できる。
4. チームの一員として作業し、リーダーシップを発揮し、情報の収集・活用を行いながら解決方法の正当性を説明し、詳しく説明できる。

発表する

「発表する」ステージでは、適切な語彙を用いて、詳細を適度に交えて自分の解決策を説明できるようにします。

1. 発見事実に基づいた根拠を自分の意見の説明に活かしておらず、決められた基準に従っていない。
2. 発見事実に基づいた根拠の一部が説明に活かされているが、自分の考えに対する説明が不十分である。決められた基準のほとんどに従うことができているが一部抜け落ちがある。
3. 適切な根拠を用いて自分の発見事実の正当性を説明し、決められた基準に従って発表することができる。
4. 決められた基準に従いながら、自分の発見事実を完全に述べ、適切な証拠を用いて自分の理論の正当性を説明することができる。



プログラミング的思考力を評価する

名前:

クラス:

| 問題点を分ける | 1.学習のはじめ | 2.学習の途中 | 3.学習のおわり | 4.学習終了後 | 注意事項 |
|--------------------------|---|---|---|---|------|
| |  |  |  |  | |
| 自分の言葉で問題を説明する。 | 自分の言葉で問題を説明できない。 <input type="checkbox"/> | 助けを借りれば、自分の言葉で問題を説明できる。 <input type="checkbox"/> | 自分の言葉で問題を説明できる。 <input type="checkbox"/> | 自分の言葉で問題を説明でき、問題を細かく分解し始めている。 <input type="checkbox"/> | |
| 問題の適切な解決策を見つけたかどうかを説明する。 | 成功の基準を説明できない。 <input type="checkbox"/> | 助けを借りれば、成功の基準を説明できる。 <input type="checkbox"/> | 成功の基準を説明できる。 <input type="checkbox"/> | かなり詳細に成功の基準を説明できる。 <input type="checkbox"/> | |
| 問題を細かく分解する方法を説明する。 | 問題を分解できない。 <input type="checkbox"/> | 助けを借りれば、問題を細かく分解できる。 <input type="checkbox"/> | 問題を細かく分解できる。 <input type="checkbox"/> | 問題を細かく分解して、各部分の関連性を説明できる。 <input type="checkbox"/> | |



プログラミング的思考力を評価する

名前:

クラス:

| 規則性を見出す | 1.学習のはじめ | 2.学習の途中 | 3.学習の終わり | 4.学習終了後 | 注意事項 |
|---|--|---|---|---|------|
| | | | | | |
| プログラムライブラリ(またはその他)からどのプログラムを利用したか、およびその理由を説明する。 | 利用したプログラムと利用した理由を説明できない。 <input type="checkbox"/> | 利用したプログラムを特定できる。 <input type="checkbox"/> | 利用したプログラムと利用した理由を説明できる。 <input type="checkbox"/> | 利用したプログラムと加えた修正を詳細に説明できる。 <input type="checkbox"/> | |
| パターンをどのように認識するか、または過去に見たコンセプトをどのように借用するかを観察する。 | パターンの認識も過去に見たコンセプトを再利用することもできない。 <input type="checkbox"/> | 助けを借りれば、パターンを認識するか、過去に見たコンセプトを再利用できる。 <input type="checkbox"/> | パターンを認識するか、過去に見たコンセプトを再利用できる。 <input type="checkbox"/> | パターンを認識するか、自分のコンセプトを再利用できる。 <input type="checkbox"/> | |



プログラミング的思考力を評価する

名前:

クラス:

| 論理的思考 | 1.学習のはじめ | 2.学習の途中 | 3.学習の終わり | 4.学習終了後 | 注意事項 |
|--|---|--|--|--|------|
| | | | | | |
| プログラム開発用の作業リストを作成する。 | 作業リストを作成できない。 <input type="checkbox"/> | 助けを借りれば、作業リストを作成できる。 <input type="checkbox"/> | 作業リストを作成できる。 <input type="checkbox"/> | 自作プログラムの開発に役立つ詳細な作業リストを作成できる。 <input type="checkbox"/> | |
| 自分の解決策のプログラミング方法を説明する。 | プログラミングについて説明できない。 <input type="checkbox"/> | 助けを借りれば、プログラミングについて説明できる。 <input type="checkbox"/> | プログラミングについて説明できる。 <input type="checkbox"/> | 各要素について詳しく説明しながら、プログラミングについて説明できる。 <input type="checkbox"/> | |
| 自分の解決策で利用したプログラミングの基本ルールを説明する(出力、入力、イベント、ループなど)。 | 解決策で利用したプログラミングの基本ルールを説明できない。 <input type="checkbox"/> | 助けを借りれば、解決策で利用したプログラミングの基本ルールを説明できる。 <input type="checkbox"/> | 解決策で利用したプログラミングの基本ルールを説明できる。 <input type="checkbox"/> | 解決策で利用したプログラミングの基本ルールを十分理解しながら説明できる。 <input type="checkbox"/> | |



プログラミング的思考力を評価する

名前:

クラス:

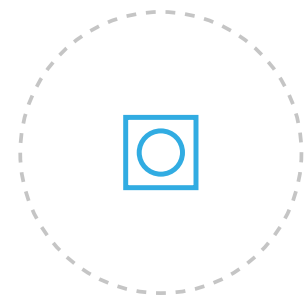


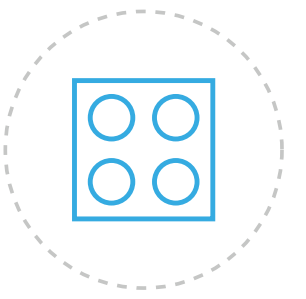
| 試行錯誤 | 1.学習のはじめ | 2.学習の途中 | 3.学習の終わり | 4.学習終了後 | 注意事項 |
|--|--|---|---|---|------|
| | | | | | |
| プログラムの実行時に起きたこととそれが自分の予想どおりだったかどうかを説明する。 | 何が起きたか説明できない。 <input type="checkbox"/> | 助けを借りれば、起きたことを説明し、予想していたことと比較できる。 <input type="checkbox"/> | 起きたことを説明し、予想していたことと比較できる。 <input type="checkbox"/> | 起きたことを説明し、予想していたことと比較でき、解決策も発見済みである。 <input type="checkbox"/> | |
| プログラムの問題をどのように修正したかを説明する。 | 問題をどのように修正したかを説明できない。 <input type="checkbox"/> | 助けを借りれば、問題をどのように修正したかを説明できる。 <input type="checkbox"/> | 問題をどのように修正したかを説明できる。 <input type="checkbox"/> | 問題をどのように修正したかをかなり詳しく説明できる。 <input type="checkbox"/> | |
| 自分の解決策と問題との関連性を説明する。 | 自分の解決策と問題との関連性を説明できない。 <input type="checkbox"/> | 助けを借りれば、自分の解決策と問題との関連性を説明できる。 <input type="checkbox"/> | 自分の解決策と問題との関連性を説明できる。 <input type="checkbox"/> | 自分の解決策と問題との関連性をかなり詳しく説明できる。 <input type="checkbox"/> | |
| プロジェクト中に新しい問題解決策をどのように試したかを説明する。 | プロジェクト中に試したほかの方法を説明できない。 <input type="checkbox"/> | 助けを借りれば、プロジェクト中に試したほかの方法を説明できる。 <input type="checkbox"/> | プロジェクト中に試したほかの方法を説明できる。 <input type="checkbox"/> | プロジェクト中に試したほかの方法を説明し、それらを解決策として検討しなかった理由をそれぞれ説明できる。 <input type="checkbox"/> | |



プログラミング的思考力を評価する

名前:

クラス:

| 発表・プレゼンテーション | 1.学習のはじめ | 2.学習の途中 | 3.学習の終わり | 4.学習終了後 | 注意事項 |
|---------------------------|---|---|---|---|------|
| |  |  |  |  | |
| 自分の解決方法で最も重要な部分を説明する。 | 自分の解決策について説明できない。 <input type="checkbox"/> | 助けを借りれば、自分の解決策について説明できる。 <input type="checkbox"/> | 自分の解決策について説明できる。 <input type="checkbox"/> | 最も重要な部分に焦点を当てて、自分の解決策について説明できる。 <input type="checkbox"/> | |
| 自分の解決策で最も重要な詳細を説明する。 | 自分の解決策で最も重要な詳細について説明できない。 <input type="checkbox"/> | 助けを借りれば、自分の解決策で最も重要な細部について説明できる。 <input type="checkbox"/> | 自分の解決策の細部について説明できるが、一部は不要である。 <input type="checkbox"/> | 自分の解決策の最も重要な詳細について説明できる。 <input type="checkbox"/> | |
| 解決策が当初の基準をどのように満たすかを説明する。 | 自分の解決策が当初の基準をどのように満たすかを説明できない。 <input type="checkbox"/> | 助けを借りれば、自分の解決策が当初の基準をどのように満たすかを説明できる。 <input type="checkbox"/> | 自分の解決策が当初の基準をどのように満たすかを説明できる。 <input type="checkbox"/> | 自分の解決策が当初の基準をどのように満たすかをきわめて明確に説明できる。 <input type="checkbox"/> | |

レゴエデュケーション® WeDo 2.0



LEGOeducation.com

LEGO and the LEGO logo are trademarks of the/son des marques de commerce du/son marcas registradas de LEGO Group.
©2017 The LEGO Group. 2017.01.01. - VI.

LEGO education