

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。




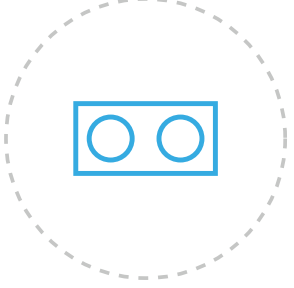
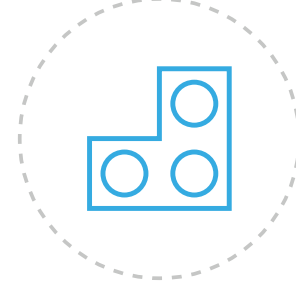
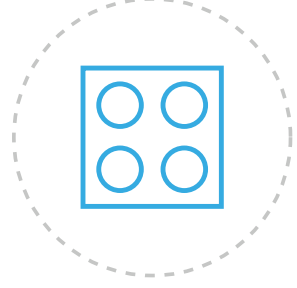


実例記録シート

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。

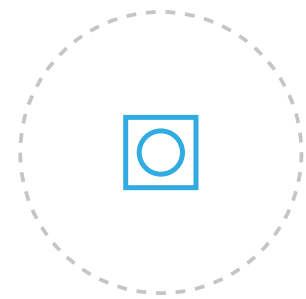


1. 発見事実に基づいた根拠を自分の意見の説明に活かしておらず、決められた基準に従っていない。
2. 発見事実に基づいた根拠の一部が説明に活かされているが、自分の考えに対する説明が不十分である。決められた基準のほとんどに従うことができているが一部抜け落ちがある。
3. 適切な根拠を用いて自分の発見事実の正当性を説明し、決められた基準に従って発表することができる。
4. 決められた基準に従いながら、自分の発見事実を完全に述べ、適切な証拠を用いて自分の理論の正当性を説明することができる。



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.学習のおわり	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:

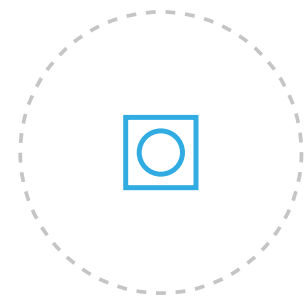


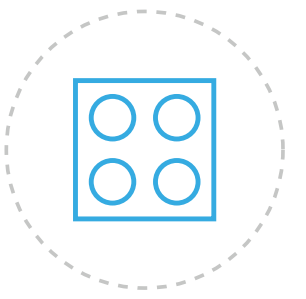
試行錯誤	1.学習のはじめ	2.学習の途中	3.学習の終わり	4.学習終了後	注意事項
プログラムの実行時に起きたこととそれが自分の予想どおりだったかどうかを説明する。	何が起きたか説明できない。 <input type="checkbox"/>	助けを借りれば、起きたことを説明し、予想していたことと比較できる。 <input type="checkbox"/>	起きたことを説明し、予想していたことと比較できる。 <input type="checkbox"/>	起きたことを説明し、予想していたことと比較でき、解決策も発見済みである。 <input type="checkbox"/>	
プログ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.学習のおわり	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詳細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"/>	