

「Scratch を卒業するタイミングについて」

はじめに

Scratch は、子どもにとって夢のような教材です。

「プログラミングは楽しい」

「自分の作品がすぐ動く」

実際、多くの子どもが「もっとやりたい！」と目を輝かせながら作品づくりに取り組んでいます。

しかし一方で、私たちは最近、

「Scratch からどのように卒業するか？」

について深く考えるようになりました。

Scratch の素晴らしいところと問題点について。

Scratch の「素晴らしいところ」については、よく知られています。

例えば、下記のようなことがあります。

- ①視覚的で直感的な操作が可能（ブロックを組み合わせることでプログラミングできます）
- ②初心者（年少者）でも簡単に動く作品が作れる
- ③ゲーム・物語・アニメーションなど多様なプログラミングができる
- ④子どもが自信を持ちやすく、継続しやすい
- ⑤どこでも、無料で利用できる

しかし、Scratch の根本的な問題については、十分に検討されていないように感じています。

特に私たちが強く感じていることは、

Scratch では、「エラーが発生しない」ことが最大の問題である
という点です。

プログラミング学習では、エラーの発生が本当の学びの入り口です。

本来、プログラミングには「間違い（バグ）」がつきものです。

プログラムを打ち間違えたり、命令の順番を間違えたり、思った通りに動かないことが日常茶飯事です。

でも、それこそが「学びのきっかけ」となります。

- ・なぜ動かないのか？
- ・何を直せばよいのか？

を考えることこそ、論理的思考力や問題解決力を育てる最大のチャンスです。

Scratch では、ブロックを自由に組み合わせるだけで作品が完成し、「エラーに出会うことがない」設計になっています。

これは一見ありがたい仕組みです。

しかし、見方を変えれば、

「失敗しないから、学びのきっかけがなく考える力が育ちにくい」ということにつながります。

Scratch はゴールではなく「プログラミング学習の入り口」と考えています。

私たちの教室では、Scratch を否定する考え方はありません。

むしろ、プログラミング学習の“入り口”としてとても優れた教材だと考えています。

しかし同時に、どのようにして Scratch を卒業するか、がとても大切だと感じています。

Scratch を使ったプログラミング学習は、例えば「自転車の補助輪付き走行」と言えます。

一定の練習を終えたら、少しずつ補助輪を外していくように、私たちは Scratch 学習に続く“ステップ”として、次のようなカリキュラムを用意しています。

- ①実世界とつながる自動車ロボットプログラミング (Tiny:bit)
- ②文字で命令を伝える Python プログラミング

- ③感性と論理をデザインとして融合する Canva
- ④本格的なゲーム制作をする Unity Visual Scripting
- ⑤データ分析をする Excel

おわりに

Scratch は素晴らしいプログラミング学習の入り口ですが、Scratch に留まりすぎると、「楽しいけど、深く問わない」学びになってしまい可能性があります。

だからこそ、私たちは「Scratch からの卒業」を大切に考えています。

本当の学びは、“うまくいかない時” にこそ、始まります。

保護者の皆さんとこの想いを共有しながら、子どもたちの成長を一緒に支えていけたら嬉しく思います。

2025 年 6 月 7 日
子どものプログラミング教室

改版履歴

日付	版数	備考
2025/06/07	初版	