

# Adapting the curriculum for pupils with SEND

The Teach Computing Curriculum has been written to support all pupils, with units containing a number of scaffolding activities and utilising effective pedagogies to ensure high quality teaching. However, you may still need to adapt resources to enable some of your pupils, for example those with special educational needs and disabilities (SEND), to access lessons fully.

The following principles will help you make adaptations that benefit all learners, and these will be more effective if you identify clearly what it is your individual pupils need help with - do they have poor working memory that means that following instructions is more difficult, or do they need help to stay focussed when completing projects?

## 1. Identify essential learning and misconceptions:

Determine the key learning in each unit that every child should know. Provide repeated opportunities for pupils to revisit this content in different ways. Identify any likely misconceptions and address these explicitly in lessons. For example, in the year 3 Animation unit, pupils tend to move characters too far between frames, so ensure this is highlighted and modelled well.

**2. Pre-teach key vocabulary:** Pre-teach the essential vocabulary for each unit, provide learners with a word list supported by images and use the vocabulary regularly throughout the unit with a consistent definition. Concentrate on a small number of terms and consider using a graphic organiser to highlight relationships between concepts, e.g. [the Frayer model](#).

**3. Create step-by-step instructions:** Break down complex tasks and routine skills for using software and hardware into smaller steps and create pictorial instructions for children to follow. For example, in year 4 Audio Editing, you could create a handout with a sequence of instructions for trimming audio clips in Audacity based on the video guide.

**4. Provide templates:** In Creating Media or Data & Information units, support task completion by providing a template for pupils to modify – removing the fear of the blank page and helping to build confidence. For example, in the year 6 Web Page Creation unit, you could set up a simple site with pages and navigation for pupils to fill in the content.

**5. Consider non-computing barriers:** Consider if difficulties in other areas, such as writing or maths, present barriers to completing a task and if so, modify the task to help mitigate these. For example, in the year 6 Communication and Collaboration unit, during lesson 3 where pupils work collaboratively to create slides, they could dictate content into the document rather than type it.

**6. Use the PRIMM framework or Parson's problems:** In programming units, add extra scaffolding using [PRIMM](#) and Parson's problems. Some pupils may not be able to create a program, but they can practise reading and exploring code in a working program, then modify it to make it more personalised. For example, in the year 4 Repetition in Games unit, for the final task learners can modify the bat catching game by changing the backdrop, adding a new costume to the sprites to change their appearance, and adding a different sound. The aim is to remove these scaffolds as children develop their skills, but some learners may not become fully independent.

**7. Harness pupils' special interests:** Increase engagement and make learning more relevant by incorporating pupils' special interests. This is also important in terms of [culturally relevant pedagogy](#). For example, in the year 5 Vector Drawing unit, pupils could use what they have learnt to create a logo for their favourite sports team.

**8. Use unplugged activities and the semantic wave:** We can use unplugged activities to help make computing concepts more relevant and understandable for learners. However, it is very important to 'repack' the knowledge of the abstract concept so that learners understand what it means in a wider context and they can use the technical language. For example, in the year 5 Selection in Physical Computing unit pupils consider examples of selection in everyday life. Children then need to see the link between these examples and how this is used in a program and have the opportunity to use the key language in context. Pupils with SEND may need repeated examples and smaller steps to repack the knowledge, e.g. you could provide learners with some printed selection blocks from the Crumble software to add everyday examples to.

**9. Support planning:** Break down the planning process into smaller parts which can be ticked off as each one is completed, and provide a planning scaffold for learners where required. For example, when planning out algorithms for Scratch, provide printed versions of the blocks to manipulate and order, to help pupils to focus on only the code required

**10. Reinforce digital skills:** A significant barrier to accessing the whole computing curriculum is a lack of key digital skills, for example being able to log on to a computer and use the keyboard effectively. Time spent revisiting digital skills across all units is important to develop fluency. Some pupils may also benefit from extra time to practise these skills in small groups, or may need image-supported help sheets to support specific repeated tasks, such as saving work.

It is important that your adaptations are informed by effective formative assessment to identify any gaps in learning and the approach which may support with these.

A further resource which can support you is the [Universal Design for Learning Framework](#) from CAST which outlines a number of approaches to include all learners in lessons by providing multiple means of engagement, representation, action and expression.

For support from fellow teachers with individual units, head over to the [STEM community](#) and join the discussions about adapting units for pupils with SEND.

Finally, there is CPD available to support you further. Complete this online course to improve your knowledge: [Creating an Inclusive Classroom: Approaches to Supporting Learners with SEND in Computing](#) or attend the face-to-face course: [Inclusive Computing in Primary Schools](#).