

创意编程，从定义上来看，是基于规则的、生成式的过程。主要是通过编写程序来制定规则，并根据这个规则来创造出具有创意和艺术感染力的作品。交互设计、信息可视化和生成艺术的内涵和外延都是创意编程的不同形式。因此，在我看来创意编程是综合性强的且兼具理性与不可控性的艺术手法。

创意编程的综合性强不仅在于它涵盖了多种设计领域、可以在多方面得到运用，也在于灵感来源范围广和表达形式的多样化。自然中植物的生长规律、鸟群的飞行轨迹，海潮的涌动，人文中舞者舞动的姿态、古典油画的改造和传统图案的衍生，都可作为创意编程的素材和灵感。由此观之，学好创意编程需要有观察发现美的眼睛。表达形式上，平面二维作品、动态图作品，三维的艺术装置、家具和服饰，甚至于声音的具象化，都可以是创意编程的表达形式。所以，在严谨中实现多元与自由成为我对创意编程最直观的印象。

创意编程的理性和不可控都围绕着创意编程的原则——基于规则的、生成式的过程。理性在于编写代码是实现艺术的手段，过程中用到数学和计算机相关知识。不可控则在于执行代码的过程中程序产生新的东西，而输出的图案、形式和行为等没有被编程，这也正是创意编程不同于传统艺术的有趣之处。

生成艺术是创意编程的重要形式，不同于“编程”这一人机合作的方法，生成艺术历史更悠久，构建规则的方式更多。同时，虽然创意编程包含程序运作，但主导为人类想法，产生具有美学意义的作品，亦可以称之为艺术

作为学生，我认为学习创意编程可以激发我们的创造力、感知力和行动力，扩展知识面，同时会运用编程为后续的设计学习提供有力工具。在人类发展中，随着技术的进步，我相信创意编程所建构出的模型会逐渐能够被建造出来，也会在各种领域发挥越来越重要的作用。