

Cubical Type Theory

Theory of Truth, Types, Equality, Identity, Universes, Univalence, HITs, Kan Filling, Cubical Type Theory

The related trinity of type theory, mathematical logic/proof theory and category theory forms a modern foundation for all of mathematics and computation.

A type is a precise mathematical specification of behavior. An element of a type satisfies its specification. Both a type and its elements are programs, subject to evaluation. Elements evaluate down to values and types evaluate down to canonical types [Martin-Löf] or type values [Harper] (same idea). Application developers experienced with mainstream programming languages will already understand elements (objects) are subject

to evaluation, but will find types being so to be new and significantly more expressive. Type theory supports the idea of indexed families of types-the index being a value (also known as dependent types). This will also be a new idea to most developers. By introducing these ideas and lots more, modern type theory goes far beyond what we see as type systems for regular programming languages. This course focuses on the most promising of the latest type theories, known as cubical type theory- a computation-friendly approach to new ideas such as univalence & higher inductive types. Open source [Agda/Cubical](#) is a good implementation.

Contents of One-Day Training Course	
<p style="text-align: center;">Target Audience</p> <p>This course is aimed at modern developers who are specifically interested in modern type theory and how it can be practically used in computation</p> <p style="text-align: center;">Prerequisites</p> <p>Good foundational mathematical education along with some programming experience, as we include exploring type theory from a computational viewpoint. Attendees can select which programming language they wish to use, as all concepts will be developed from first principles.</p>	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p style="text-align: center;">A New Foundation</p> <p>Type theory Category theory Mathematical logic/proof theory How they relate to each other (Shulman) Constructivism</p> <p style="text-align: center;">Type Theory Overview</p> <p>Think of modern type theory as a highly expressive programming language useful for mathematics (+everything based on it) A theory of truth – based on Martin-Löf's ground-breaking paper: "Constructive mathematics and computer programming"</p> <p style="text-align: center;">Basic Concepts</p> <p>Type Element Evaluation Judgment Witness Intuitionistic Type Theory</p> <p style="text-align: center;">Simple Types</p> <p>Defining a type Defining an element Look at bool and nat(ural number)</p> <p style="text-align: center;">Common Types</p> <p>Type Function Sum Product Top / Bottom</p> <p style="text-align: center;">Dependent Types</p> <p>The idea behind dependent types Creating an indexed families of types</p> </div> <div style="width: 45%;"> <p style="text-align: center;">Additional Concepts</p> <p>The meaning explanation Functionality (in a type theory sense) The hypothetical Recursion</p> <p style="text-align: center;">Equality and Identity</p> <p>Introduction to equality in type theory Variations of equality type (exact, ..) Identifications</p> <p style="text-align: center;">Higher Inductive Types (HITs)</p> <p>HITs as generalization of inductive types Uses in various forms of construction Diagonals</p> <p style="text-align: center;">Univalence</p> <p>Initially considered in terms of Homotopy Type Theory (HoTT) That uses an axiom – not desirable from a computational viewpoint – why? So other approaches explored ...</p> <p style="text-align: center;">Cubical Type Theory - Intro</p> <p>This is the cutting edge (2019) type theory Simple intro Uses points, lines, planes, cube, n-cube The idea of paths Transporting</p> <p style="text-align: center;">Cubical Type Theory - Details</p> <p>Kan filling Higher dimensionality Cartesian variant</p> <p style="text-align: center;">Adga/Cubical</p> <p>Agda/Cubical is an open source implementation of cubical type theory – let's explore how it works</p> </div> </div>