

# Tutorial 1: Getting Started with STUNIR

---

**Duration:** 10 minutes

**Level:** Beginner

**Prerequisites:** Python 3.8+, terminal access

---

## Video Script

---

### Introduction (0:00 - 0:30)

“Welcome to STUNIR! In this tutorial, you’ll learn how to install STUNIR and generate your first deterministic receipt. By the end, you’ll understand the core concepts and be ready to explore more advanced features.”

### What is STUNIR? (0:30 - 1:30)

“STUNIR stands for **S**pec **T**o **U**niversal **I**ntermediate **R**epresentation. It’s a toolkit for generating deterministic, verifiable code from specifications.”

#### Key points to cover:

- Deterministic output (same input = same output)
- Multi-language support (Python, Rust, C, Haskell)
- Cryptographic verification via receipts
- Reproducible builds

### Installation (1:30 - 3:00)

#### Terminal Demo:

```
# Clone the repository
git clone https://github.com/stunir/stunir.git
cd stunir

# Check Python version
python3 --version # Should be 3.8+

# Install dependencies (optional, for full features)
pip install -r requirements.txt

# Verify installation
python3 -c "import json, hashlib; print('Ready!')"
```

“STUNIR is self-contained - the core functionality uses only Python standard library. Optional dependencies enable advanced features.”

### Your First Spec (3:00 - 5:00)

Create `my_first_spec.json` :

```
{
  "name": "hello_module",
  "version": "1.0.0",
  "functions": [
    {
      "name": "greet",
      "params": [{"name": "name", "type": "str"}],
      "returns": "str"
    }
  ],
  "exports": ["greet"]
}
```

“A spec defines your module’s structure - name, version, functions, and exports. This is the input to STUNIR.”

## Generating IR (5:00 - 7:00)

### Terminal Demo:

```
# Generate IR from spec
python3 tools/ir_emitter/emit_ir.py my_first_spec.json output/hello.ir.json

# View the generated IR
cat output/hello.ir.json
```

### Expected output:

```
{
  "ir_version": "1.0.0",
  "ir_epoch": 1738000000,
  "ir_spec_hash": "abc123...",
  "module": {"name": "hello_module", "version": "1.0.0"},
  "functions": [...]
```

“The IR includes a hash of your spec and a timestamp (epoch). This ensures traceability.”

## Verifying Determinism (7:00 - 8:30)

### Terminal Demo:

```
# Run generation multiple times
for i in 1 2 3; do
  python3 tools/ir_emitter/emit_ir.py my_first_spec.json /tmp/test_${i}.json
  sha256sum /tmp/test_${i}.json
done
```

“Notice the hashes are identical! This is determinism - the cornerstone of STUNIR.”

## Generating a Receipt (8:30 - 9:30)

### Explain receipts:

“Receipts are cryptographic proofs of your build artifacts. They enable verification without rebuilding.”

```
# Generate manifest and receipt
python3 manifests/ir/gen_ir_manifest.py --ir-dir output/

# View the receipt
cat receipts/ir_manifest.json
```

## Wrap Up (9:30 - 10:00)

### Summary:

- ☒ Installed STUNIR
- ☒ Created a spec
- ☒ Generated IR
- ☒ Verified determinism
- ☒ Created a receipt

### Next steps:

“In the next tutorial, we’ll explore the complete workflow including multi-target code generation. See you there!”

---

## Commands Summary

```
# Installation
git clone https://github.com/stunir/stunir.git
cd stunir

# Generate IR
python3 tools/ir_emitter/emit_ir.py <spec.json> <output.json>

# Generate manifest
python3 manifests/ir/gen_ir_manifest.py --ir-dir <dir>

# Verify manifest
python3 manifests/ir/verify_ir_manifest.py receipts/ir_manifest.json
```

## Troubleshooting

If you encounter issues:

1. Check Python version: `python3 --version`
2. Verify JSON syntax: `python3 -m json.tool < spec.json`
3. See [Troubleshooting Tutorial](#) (04\_troubleshooting.md)